DOT/FAA/AR-00/17

Office of Aviation Research
Washington, D.C. 20591

Denver International Airport Sensor Processing and Database

March 2000

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/AR-00/17		

4. Title and Subtitle	5. Report Date
DENVER INTERNATIONAL AIRPORT SENSOR PROCESSING AND DATABASE	March 2000
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Mingyao Dong and Gordon F. Hayhoe	
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Galaxy Scientific Corporation 2500 English Creek Ave., Bldg. C Egg Harbor Twp., NJ 08234-5562	11. Contract or Grant No. DTFA030-95-D-00019
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Federal Aviation Administration Office of Aviation Research Washington, DC 20591	Final Report 14. Sponsoring Agency Code AAS-200

15. Supplementary Notes

The Federal Aviation Administration William J. Hughes Technical Center COTR was Dr. Satish K. Agrawal

16. Abstract

Data processing and database design is described for an instrumentation system installed on runway 34R at Denver International Airport (DIA). Static (low-speed) and dynamic (high-speed) sensors are installed in the pavement. The static sensors include thermistors, resistivity probes, time domain reflectometer (TDR) moisture gages, strain gages, and joint gages sampled at a rate of once per hour. Dynamic sensors include infrared (IR) position sensors, multidepth deflectometers, strain gages, and geophones sampled at approximately 160 Hz.

Data processing is performed on a personal computer and consists of reading the raw data files, conversion to engineering units, formatting for storage in the database, and conversion to Structural Query Language (SQL) statements for loading the database. Peak values in the dynamic sensor records are automatically computed. The database is implemented in Oracle on a Unix-based workstation. The data structure for storing data from the dynamic sensors is based on single aircraft events consisting of all data collected during the passage of a single aircraft over the section of instrumented pavement.

17. Key Words	18. Distribution Statement
Full-scale pavement testing, Pavement instrumentation, Airport pavement response	Document is available to the public through the National Technical Information Service (NTIS) Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	54	

Form DOT F1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

A section of Runway 34R at Denver International Airport was instrumented during construction of the runway. The instrumented section is located close to the runway threshold, which is close to where the aircraft begin their takeoff roll. The runway is of rigid construction and has more than 300 sensors embedded in the structural layers. Low-speed measurements are taken once per hour of the temperature, moisture, concrete strain, and joint movement. High-speed measurements, at a rate of 100 per second, are taken of the concrete strain and vertical movement of the structural layers. The low-speed measurements are taken continuously. The high-speed measurements are taken only when an aircraft passes over the instrumented pavement. Infrared light sensors beamed across the runway are used to trigger the high-speed data collection and to sense the velocity and position of the aircraft.

Data collected at the site is stored in raw binary format and then transmitted to the William J. Hughes Technical Center via a high-speed data link. After transmission, the files are processed to convert the data to engineering units and to condense it into a form suitable for storage in an on-line database. Processing of the high-speed data includes the identification of the maximum value recorded for each sensor during the passage of each aircraft. All of the maximum values above a certain threshold are stored in the database. Ten percent of the sensor responses in each sensor group which have the largest maximum values are also stored in the database as complete data records. The low-speed data is processed separately and stored complete in the database. Additional information, such as pavement design and construction information, pavement evaluation survey results, and falling weight deflectometer (FWD) data, is also stored in the database.

The database is implemented in Oracle on a dedicated workstation and is accessible on the Internet. The data can be accessed through a custom interface or directly with queries submitted in the Structured Query Language (SQL).

1. INTRODUCTION.

The Airport Technology R&D Branch at the Federal Aviation Administration William J. Hughes Technical Center has instrumented a section of runway 34R at the new Denver International Airport (DIA). The instrumentation consists of dynamic sensors embedded in the pavement structure which are sampled at approximately 160 Hz and static sensors which are sampled at rates of once per hour or less.

Dynamic sensor data are recorded for a period of approximately 8 seconds while an aircraft passes over the instrumented section of pavement. The raw data records are stored in a binary data (.DAT) file. Information on the format and contents of the .DAT files is stored in signal (.SIG) files and information on the recording parameters is stored in playback (.PLB) files. All of the files for a given aircraft event are stored on-site on an IBM compatible computer and transmitted electronically to the Technical Center for processing and analysis.

The static sensor data primarily consists of temperature, moisture, and strain measurements from sensors embedded in the pavement structure and recorded and stored in ASCII files by a data logger. Weather station data are also recorded in ASCII files.

In total, there are over 300 sensors in the instrumented pavement. All of the data from the sensors must be reduced and made available for analysis in an easily accessible form. To do this, the Technical Center has developed an Oracle database for storing relevant data from the pavement sensors. The information stored in the database allows for long-term analysis of the behavior of the pavement and for analysis of pavement response to many different aircraft under different environmental conditions.

Computer programs and procedures developed for processing the dynamic sensor data are described in section 2. The primary objectives of the dynamic sensor data processing were to identify those sensors giving significant response during passage of an aircraft, to compute values of peak response for the identified sensors, and to provide a straightforward means of selecting records of limited length for storing the time histories of significant events without consuming large amounts of storage space. The final step in the processing is to apply the correct calibrations and prepare Structured Query Language (SQL) statements for loading the data into the database.

Computer programs and procedures developed for processing the static sensor data are described in section 3. The processing consists mainly of applying the correct calibrations, converting to a consistent set of units, and preparing SQL statements for loading the data into the database.

The Oracle database was implemented on an SGI Indigo2 workstation. The structure of the database is described in section 5.

Detailed information about the placement and characteristics of the sensors is only given where it is necessary in the description of the data processing. Complete information on the sensors is stored in the database or can be found in other sources available from the Technical Center.

2. DYNAMIC SENSOR PROCESSING.

The dynamic sensor signals are processed with a Visual Basic computer program which reads the data files for a selected aircraft pass, displays the data, and provides functions for saving selected data in an SQL file for transfer to the Oracle database.

The user window for the program is shown figure 1. The sensors are organized in tabbed groups, with the H-bar strain gage group shown selected. At the top left of each tab is a display of the selected data files. Below the file display are text boxes for the plot parameters. At the top right is a grid box showing summary information for individual sensors in the selected group. In the center is a plot of the selected sensor signal drawn according to the settings of the plot parameters. At the bottom left is a group of command buttons for operating on the sensor signals. At the bottom right are two text boxes, one gives brief instructions and the other gives the status of the current view or operation. The date and time of the test are displayed in the window title bar in Year/Julian Day/Hour/Minute format.

Each section of the user window is now described in sufficient detail for operating the program and interpreting the results.

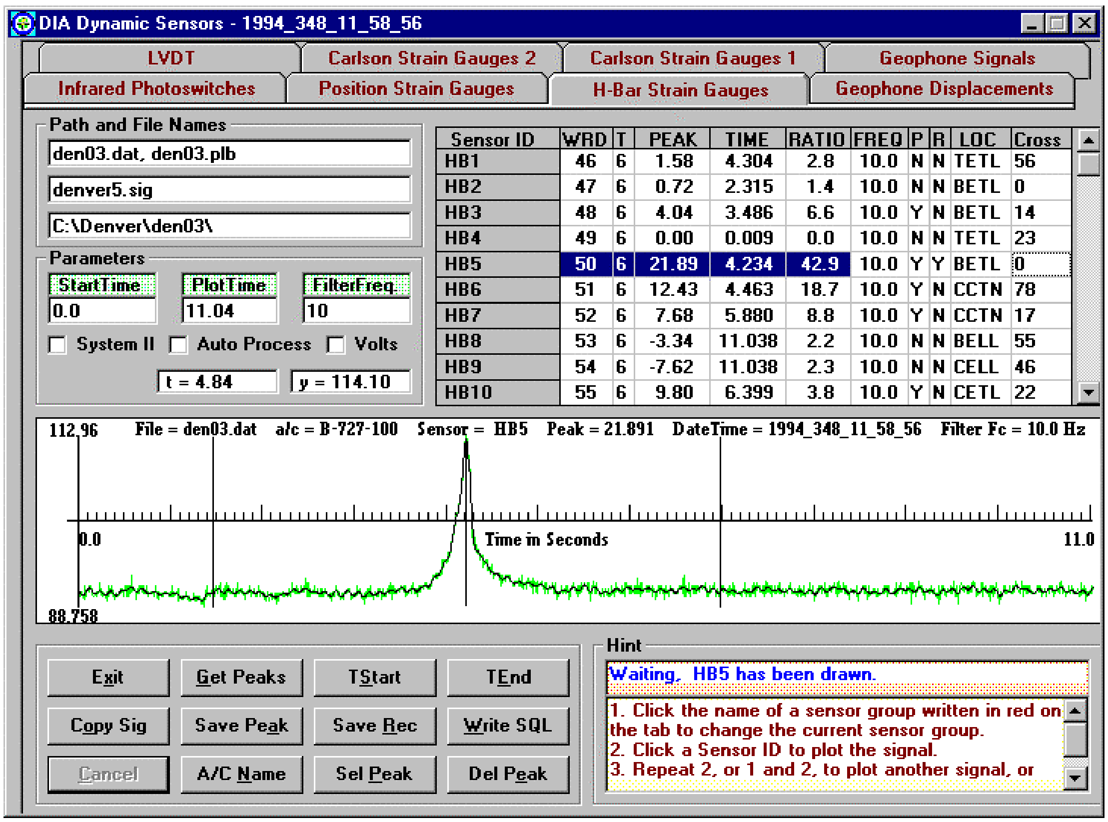


FIGURE 1. USER WINDOW FOR SENSOR SELECTION, RECORD DISPLAY, AND FUNCTION EXECUTION

2.1 OPENING A FILE.

Clicking the mouse within one of the file text boxes at the top left displays the file selection dialog shown in figure 2. The file path is selected from the Drive List and Directory List boxes and the file is selected from the File List box. If the .DAT and .PLB box is selected to open the file selection dialog, the File List first displays all .DAT files in the selected path. After selecting one of the .DAT files, all .SIG files in the selected path are displayed. Selecting one of the .SIG files then closes the file dialog, returns control to the user window, and loads and processes the file. During loading, all of the sensor data are stored in memory as a 2-dimensional array and all of the .SIG file data are stored in arrays in memory. During processing, all of the sensor data are converted to voltages, according to the signal type, and converted to engineering units according to calibration constants initialized on program startup.

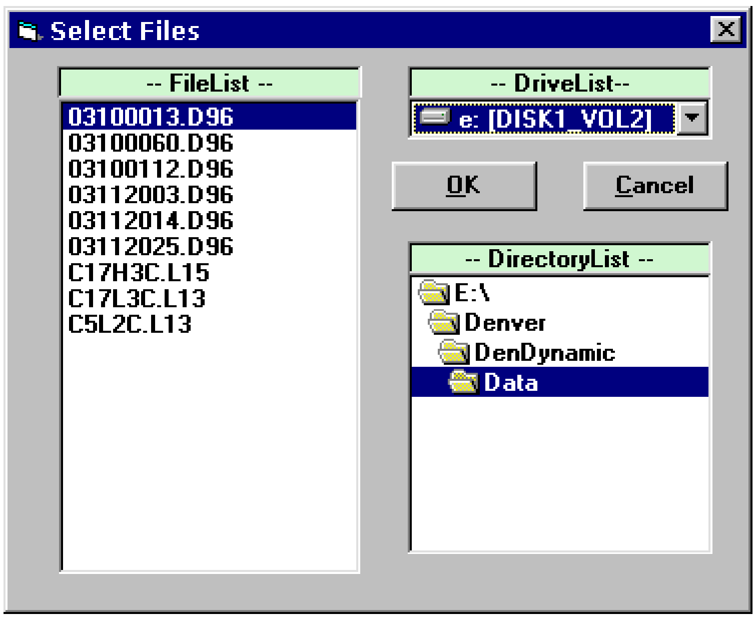


FIGURE 2. FILE SELECTION

If synchronization is lost in the .DAT file, the number of frames lost is estimated and the signal values in those frames is set to the values read in the last valid frame.

2.2 ACTIVATING AND PLOTTING SENSOR SIGNALS.

After a file has been loaded and processed, a sensor group must be selected by clicking the mouse on one of the tabs. The sensor names are then listed in the grid box in the same order as they are listed in the .SIG file. Clicking on one of the sensors in the grid plots the raw and filtered signals and finds the primary peak in the signal (see below). Information in the grid is as follows:

Sensor ID. The name of the sensor as stored in the .SIG file. The only exception is the Carlson 2 sensors, which have the 2 removed for compatibility with the database table names.

WRD. The WORD number of the sensor as listed in the .SIG file. These numbers are used as internal references and are not used for identification in the database.

T. The WORD type as stored in the .DAT file.

4 = 0 to 5 volts
6 = -5 to +5 volts
7 = -10 to + 10 volts
8 = linear with coefficients stored in .SIG file

PEAK. The value of the primary peak in the units for the selected signal.

TIME. The time at which the primary peak occurs in the data record.

RATIO. The ratio of the value of the primary peak to the root mean square (rms) estimate of the noise in the signal.

FREQ. The low-pass filter cutoff frequency at which the peak and ratio values were computed.

P. Peak values are selected for storage in the database if set to "Y." Not selected for storage if set to "N."

R. The peak record is selected for storage in the database if set to "Y." Not selected for storage if set to "N."

LOC (only for H-Bar). Location code for sensor placement (described later).

Cross (only for H-Bar). Cross reference for sensors placed at the same horizontal position but at different depths. This column is displayed by scrolling the grid with the arrow keys on the keypad.

2.3 PLOT FORMAT AND PRIMARY PEAK CALCULATION.

The sensor plot range is changed by changing the settings in the Start Time and Plot Time Parameter boxes (units are seconds). Start Time is the time at which the plot starts within the complete data record. Plot Time is the duration of the plot. For example, if Start Time is 1.5 and Plot Time is 2.5, the time axis on the plot will start at 1.5 seconds and end at 4.0 seconds.

The green (light-shaded gray-scale) plot is the raw signal as recorded in the .SIG file except that it has been converted to engineering units. The sample rate is approximately 160 Hz and the bandwidth of the raw signal is therefore approximately 80 Hz.

The black (smooth) plot is a low-pass filtered version of the raw sensor signal. A fourth order tangent (Butterworth) IIR digital filter is used with forward and backward passes to eliminate filter phase shift. The total filtering operation is therefore eighth order. The cutoff frequency of the filter is selected, in Hz, in the Parameters Filter Freq box.

When a sensor signal is plotted, the primary peak in the signal is also found using the following procedure:

a. Filter the signal with the currently selected filter cutoff frequency.

b. Find the minimum and maximum values in the filtered signal record.

c. Find the average value of the filtered signal record.

d. Find the absolute values of the differences between the minimum and maximum values and the average value. The largest of the absolute values gives the direction of the peak. For example, if (average-minimum) is larger than (maximum-average) then the peak is at the minimum value.

e. Set start and end times for the peak record. By default, the start time is set at one quarter the signal record length to the left of the time for the peak, and the end time is set at one quarter the signal record length to the right. The start and end times can be changed as described later.

f. Find the left offset by computing the average value of the filtered signal between the beginning of the signal record and the start time of the peak record. If the number of samples between the two points is less than 20, the left offset is set to a large value to signify that it is invalid.

g. Find the right offset by computing the average value of the filtered signal between the end of the signal record and the end time of the peak record. If the number of samples between the two points is less than 20, the right offset is set to a large value to signify that it is invalid.

h. If the left offset is valid, subtract the left offset value from the minimum or maximum value (whichever has been selected), otherwise subtract the right offset value from the minimum or maximum value. This gives the value of the **primary peak** for the signal record.

i. Find the root mean square (rms) of the noise in the signal. The rms is calculated by subtracting the raw signal from the filtered signal, squaring, summing, and dividing by the number of samples.

j. Find the ratio of the absolute value of the primary peak to the rms of the noise. This gives an estimate of the strength of the peak and can be used for screening sensor signals for storage in the database. For example, a ratio of 1 probably means that the selected sensor was not excited by the passage of the aircraft and the signal primarily contains noise.

k. Store the computed values for display and transfer to the database.

The peak data are stored in a VB Type structure:

```
Type PeakDataPrototype
  FilterFreq As Single         ` Cutoff frequency for Peaks data.
  LeftFraction As Single
  RightFraction As Single
  TStart As Single
  TEnd As Single
  NPeaks As Integer             ` No of peaks, maximum = 4.
  IPeak(MaxPeaks) As Integer    ` Array of peak sample numbers.
  TPeak(MaxPeaks) As Single     ` Array of peak times.
  ValPeak(MaxPeaks) As Single   ` Array of peak values.
  PeakToRMS As Single           ` Ratio, peak value to rms noise.
  Threshold As Single           ` For auto. peak save selection.
  OffsetLeft As Single          ` Left offset value.
  OffsetRight As Single         ` Right offset value.
  Selected As Integer           ` True to save peaks.
  SaveRecord As Integer         ` True to save peak record.
End Type
```

An array PeakData() of Type PeakDataPrototype is declared, with one index for each sensor.

The peak record start and end times are shown on the plot as vertical lines from the top to the bottom of the plot area. The primary peak is shown on the plot as a vertical line from the peak to the top or the bottom of the plot area.

Data for the primary peak are always stored first in the peak data field arrays IPeak(), TPeak(), and ValPeak(). Data for secondary peaks (see below) are stored in the field arrays in the same order as they are selected.

2.4 CHANGING THE PEAK RECORD START AND END TIMES.

The peak record start time is changed as follows:

a. Click the TStart command button at the bottom left of the user window.

b. Click on the plot at the position of the new start time. The start time can only be positioned to the left of the primary peak.

Changing the peak record end time is the same except that the TEnd command button is clicked first.

2.5 SELECTING AND DELETING SECONDARY PEAKS.

Up to three secondary peaks can be selected on a signal record by the following procedure:

a. Click the Sel Peak command button at the bottom left of the user window.

b. Click on the plot to the left and right of the peak to be selected. The maximum and minimum values of the filtered signal record are found between the two points and the previously computed left (or right if the left is invalid) offset value subtracted. The largest absolute value of the two is selected as the value of the secondary peak. If the value of the secondary peak is the same as the signal value at either the left or right boundaries of the selection, a peak is considered to not be present and the selection closed without storing a new peak. Also, if the secondary peak is outside the peak record the selection is closed without storing a new peak.

A secondary peak is deleted by clicking the Del Peak command button followed by clicking on the plot close to the peak to be deleted. The primary peak cannot be deleted.

2.6 MARKING SENSORS FOR DATA STORAGE IN THE DATABASE.

There are two options for storing sensor data in the database. The first is to store the peak values together with other table data such as filter frequency. The second is to additionally store the time history data values of the peak record.

The base option is selected by clicking the **Save Peaks** command button. The button is actually a toggle, and successive clicks will select/deselect the sensor data for storage. The status of the selection (Y or N) is shown in the P column of the sensor grid.

The second option is toggled in the same way with the **Save Rec** command button. Selecting the second option automatically selects the base option as well. The base option cannot be deselected while the second option is selected.

For H-Bar strain gages, up to three gages can be located at the same horizontal position in the pavement but at different depths. If one gage in a group of gages at the same horizontal position is selected for storage, the others in the group are automatically selected for storage as well.

The location and orientation of the H-Bar gages is also identified by a four-character location code. The key to the code is:

First character:

T = top of the slab
B = bottom of the slab
C = located in the cement-treated base

Second character

C = located in the center of a slab
E = located at the edge of a slab

Third character

T = when the gage is at an edge, the joint is transverse
L = when the gage is at an edge, the joint is longitudinal
T = when in the center, the gage is oriented transversely
L = when in the center, the gage is oriented longitudinally

Fourth character

P = perpendicular to the joint
L = parallel to the joint
N = neither

2.7 DEFAULT PEAK SELECTION.

The command button Get Peaks automatically runs through all linear variable differential transformers (LVDT), position strain gages, H-Bar strain gages, and Carlson strain gages and finds the peak and ratio values for each sensor. When a sensor group is selected after Get Peaks has been clicked, the peak and ratio values are displayed in the sensor grid. This allows for more rapid screening of the sensors for storage in the database.

The Get Peaks function also automatically selects sensor records for peak storage in the database based on a threshold Ratio value initialized on program startup. The threshold values are 4 for position strain gages and 3 for all other sensor types. After selecting sensor records for peak storage, the largest three ratio values in each sensor group are selected for storage of the peak records of the associated sensors.

Note: The grid display for the sensor group selected at the time Get Peaks is run is not automatically updated. To display the peak and ratio values for the selected group, select another group after running Get Peaks and then reselect the original group.

Note: The procedures for setting and saving the filter cutoff frequency during peak detection are fairly complicated. The rules are as follows:

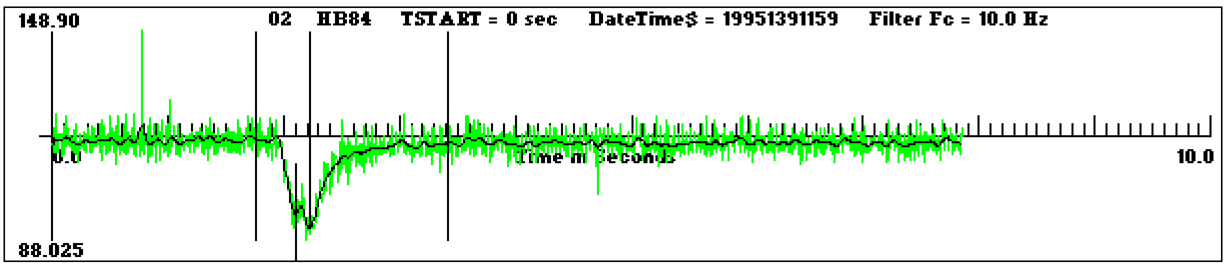
a. Each time a sensor signal is plotted, the primary peak is recomputed using the filter cutoff frequency displayed in the FilterFreq Parameter box. This filter frequency is stored in the FilterFreq field of the PeakData() array together with the peak data in the appropriate fields.

b. When a secondary peak is selected, the filtered signal shown in the plot is used to find the peak. The cutoff frequency is not stored other than in the FilterFreq field following primary peak computation. Therefore, up to three secondary peaks can be found with three different cutoff frequencies, but the frequency for each secondary peak is not stored with the peak data. Only the frequency used the last time the signal is plotted is stored.

c. When Get Peaks is run, the filter cutoff frequencies used to compute the primary peaks are those stored in the FilterFreq field. These are not necessarily the same as the frequency currently set in the FilterFreq parameter box. Plotting a signal after Get Peaks is run may therefore change the primary peak data. Default values stored in the FilterFreq field at program startup are 10 Hz for LVDT, H-Bar strain gages, Carlson strain gages, and Geophone signals, and 15 Hz for position strain gages.

2.8 COPYING A PLOT TO THE CLIPBOARD.

The command button Copy Sig writes basic information at the top of the plot and saves a bitmap of the plot to the clipboard. A sample is given below. The sample is the same as the plot shown in the full user window above except that TStart and TEnd have been moved and a secondary peak has been marked. The date format is Year, Julian Day, Hour, Minute, Second.



2.9 ENTERING THE AIRCRAFT NAME.

Clicking the A/C Name command button displays an input box for entering the name, or some other ID, to identify the aircraft type. Up to 16 characters can be used to describe the aircraft. If more than 16 characters are entered, the leftmost 16 characters are selected. If a name is not entered, “NA” is saved. The aircraft name can also be entered directly into the database table on the workstation.

2.10 AIRCRAFT PICTURE.

A reference to a bitmap of a video frame of the aircraft is also stored in the database. For the bitmap to be retrievable, it must be transferred to the workstation with the correct path and file name. The default file name is the date/time for the event, which is stored in the database automatically with the default path. If the bitmap file is transferred to a different path and name, the reference must be changed in the database table. The default path and file name can be retrieved from the database table.

2.11 SAVING TO AN SQL FILE FOR DATABASE STORAGE.

After sensors have been selected for storage of peaks and peak records into the database, an SQL file is created by clicking the command button Write SQL. The name of the file is taken from the rightmost eight characters of the date/time string for the aircraft pass (see paragraph 2.8 for the format), with .SQL appended. After creating the SQL file, it is transferred to the Oracle workstation and run to load the new data into the database. A sample file, with the name 51391159.SQL is shown below.

```
-- Load the aircraft table and increment aircraft event number.
insert into dong.aircraft
   values (dong.seq_event#.nextval,'19951391159',59,'NA',,,3,6.727,
21.398,3.082,'','/usr/oracle/dong/acpicture/19951391159.BMP');

-- Load a new event record into the table event_record for HB84, including peak record.
insert into dong.event_records
   values (dong.seq_event#.currval,'HB84','128 0.012672 -0.502 -0.001
0.564 0.537 -0.062 -0.591 -0.619 -0.412 -0.368 -0.289 0.244 0.874
0.901 0.480 0.104 -0.720 -2.948 -6.281 -9.138 -10.641 -11.601 -13.434
-16.615 -19.957 -21.781 -21.839 -21.163 -20.448 -19.616 -18.752 -
18.761 -20.459 -23.145 -25.128 -25.571 -24.960 -24.121 -23.352 -22.354
-20.867 -19.023 -17.136 -15.600 -14.509 -13.406 -11.959 -10.528 -9.895
-10.223 -10.353 -9.116 -7.057 -5.894 -6.486 -7.687 -7.594 -5.808 -
3.797 -3.212 -4.200 -5.549 -6.084 -5.461 -4.217 -3.249 -2.949 -3.139 -
3.567 -3.995 -4.163 -3.943 -3.479 -3.113 -3.038 -3.146 -3.162 -2.814 -
2.233 -1.913 -2.079 -2.514 -2.812 -2.648 -2.071 -1.601 -1.734 -2.325 -
2.657 -2.296 -1.636 -1.361 -1.573 -1.688 -1.321 -0.876 -0.770 -0.720 -
0.287 0.347 0.440 -0.416 -1.642 -2.322 -2.157 -1.614 -1.446 -1.906 -
2.471 -2.615 -2.351 -1.897 -1.316 -0.687 -0.358 -0.590 -1.226 -1.909 -
2.232 -1.962 -1.400 -1.095 -1.166 -1.269 -1.202 -1.112 -1.186 -1.374 -
1.321');

-- Load peak data into the table HB84.
insert into dong.HB84
   values
(dong.seq_event#.currval,0.006336,1.786,3.403,15.000,2,117.068,116.282
,2.218,-25.571,2.097,-21.977,0.000,0.000,0.000,0.000,'');

-- Load a new event record into the table event_records for MDD7G1, no peak record.
insert into dong.event_records
   values (dong.seq_event#.currval,'MDD7G1','');

-- Load peak data into the table MDD7G1.
insert into dong.MDD7G1
   values      (dong.seq_event#.currval,0.006336,1.386,3.992,15.000,1,-
83.901,-82.712,2.028,-12.161,0.000,0.000,0.000,0.000,0.000,0.000,'');

-- Commit all data for permanent storage in the database.
commit;
```

2.12 INFRARED SENSOR DATA PROCESSING.

The infrared (IR) sensors are IR emitters and detectors located on opposite sides of the runway. They are used to detect the passage of the aircraft landing gears and to calculate aircraft velocity and position. Normally, the detector outputs are zero volts. But when a landing gear interrupts the beam between an emitter-detector pair, the output changes to 5 volts. The first time a detector output rises above 0.5 volt is taken to be the time at which the nose gear of the aircraft interrupts the beam. IR sensors 1, 2, 4, 5, and 7 are perpendicular to the runway centerline and are used to compute the longitudinal position and velocity of the aircraft (under the assumptions that longitudinal acceleration and lateral position are constant as the aircraft passes through the beams). Sensors 3 and 6 point diagonally across the runway (1 unit longitudinally to 2 units laterally) and are used to compute, in conjunction with the perpendicular sensor signals, the lateral position of the aircraft's nose gear. Under some circumstances, the detectors are obscured due to environmental conditions and do not change voltage properly when the gear cuts the beam. Checks (described below) are therefore made during processing to eliminate those sensors which may not have triggered properly.

IR sensor 1 is also used to trigger the start of data collection and the time it triggers represents event time zero against which all other occurrences in the event are referenced.

Processing of the IR sensors is as follows:

a. Loop through all samples for each sensor and store the sample number at which each sensor voltage first rises above 0.5 volt. Convert to time and subtract the time for sensor 1. Sensors 1 and 2 are located next to each other and are assumed to trigger at the same time.

b. Eliminate each perpendicular sensor which did not trigger, keeping track of the total number of valid perpendicular sensor signals.

c. Eliminate perpendicular sensors which did not trigger in sequence. For example, eliminate sensor 4 if its trigger time is later than the trigger time for sensor 5 (this probably means that sensor 4 triggered on the main gear but not on the nose gear). Save the total number of remaining valid perpendicular sensor signal for storage in the database.

d. Fit a curve to the valid perpendicular sensor times and longitudinal positions to give a continuous equation for distance traveled versus time. For one valid signal, an equation cannot be defined. For two valid signals, a linear line through the points is used, giving an assumed constant velocity and zero acceleration. For three valid signals, a quadratic line is used, giving an assumed constant acceleration. For four valid signals, a least squares quadratic curve fit is used, giving an assumed constant acceleration. In all cases, the longitudinal position of the nose gear is given by the equation:

$$x = v_0 \times t + a \times \frac{t^2}{2}$$

where: t = time from event time zero

v_0 = slope of the curve at event time zero

a = acceleration computed from the curve fit

e. Compute the lateral position of the nose gear at the trigger times for sensors 3 and 6 using:

$$y_3 = 2 \times x_3 - 85$$
$$y_6 = 85 - 2 \times x_6$$

where: $x_{3,6}$ = trigger distance for sensors 3 and 6
$y_{3,6}$ = nose gear lateral position at trigger time for sensors 3 and 6
85 = one half the perpendicular spacing between sensors 3 and 6

f. Lateral position is assumed to be constant during travel past the sensor beams and is computed as the average of y_3 and y_6.

g. The v_0, a, and average lateral position are saved for storage in the database.

For illustration, results for the FAA B-727 passing over the pavement section are shown in figures 3 and 4.

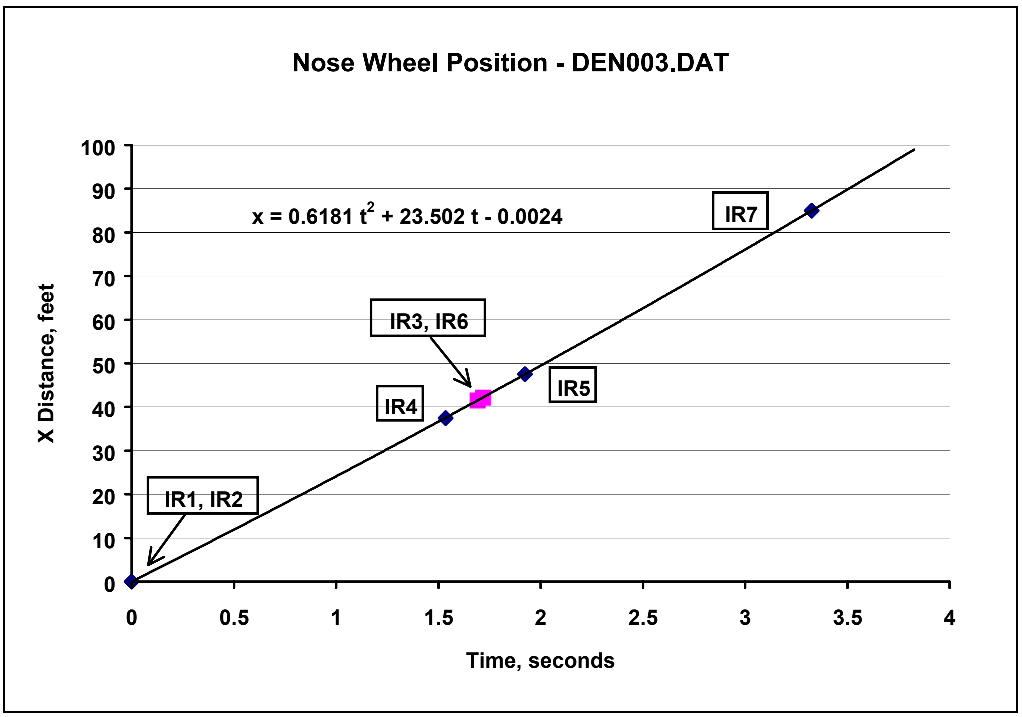


FIGURE 3. TYPICAL INFRARED SENSOR TRIGGER TIMES

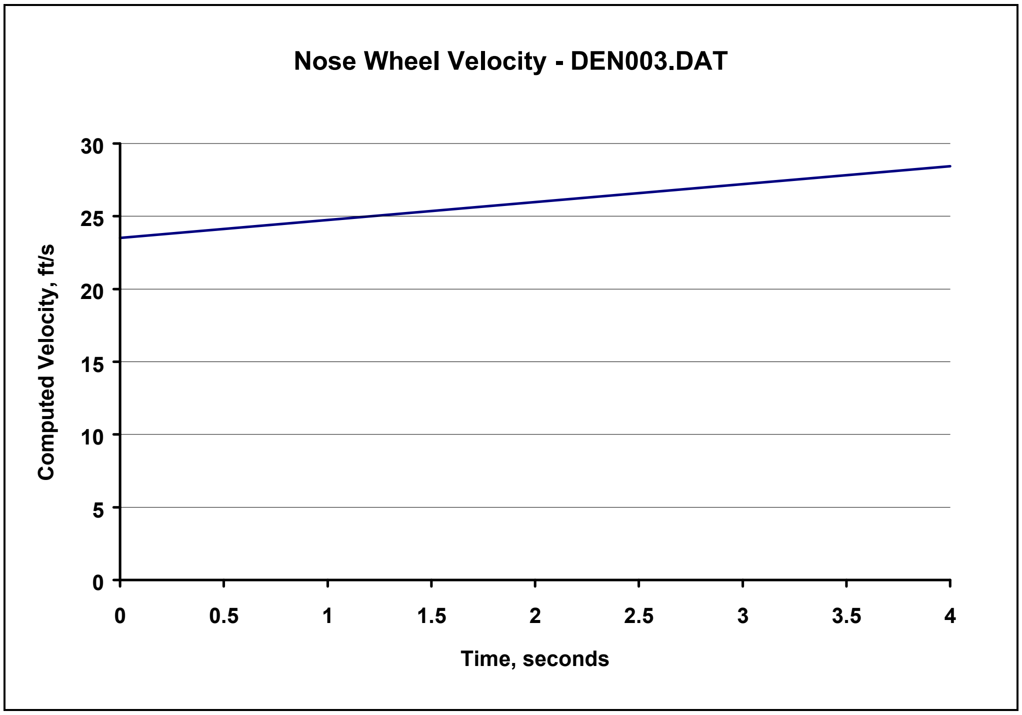


FIGURE 4. COMPUTED VELOCITY FOR TRIGGER TIMES IN FIGURE 3

2.13 LVDT DATA PROCESSING.

No processing of the LVDT signals is required other than converting to engineering units according to the calibrations supplied. The units displayed in the user window, and stored in the database, are thousandths of an inch (mils). Calibrations for the LVDTs are in the range 39 to 42 volts/inch.

2.14 POSITION STRAIN GAGE DATA PROCESSING.

No processing of the position strain gage signals is required other than converting to engineering units. The units of the signals displayed in the user window, and stored in the database, are microinches/inch. The calibration for all position strain gages is 150.23 microinches/inch/volt.

2.15 H-BAR STRAIN GAGE DATA PROCESSING.

No processing of the H-bar strain gage signals is required other than converting to engineering units. The units of the signals displayed in the user window, and stored in the database, are microinches/inch. The calibration for all H-bar strain gages is 150.23 microinches/inch/volt.

2.16 CARLSON-1 STRAIN GAGE DATA PROCESSING.

The Carlson-1 strain gage signals provide the excitation voltages applied to the gages and are required in the calculation of the Carlson gage strain outputs. Processing consists of converting the .DAT file values to volts and averaging over the full signal record for each Carlson gage.

2.17 CARLSON-2 STRAIN GAGE DATA PROCESSING.

The Carlson-2 strain gage signals provide the output voltages from the gages. Computation of the strain outputs is as follows:

$$\varepsilon = \left(\frac{1}{0.5 - \dfrac{V_2}{1{,}000 \times V_1}} - R_{REF} \right) \times Cal \times 10{,}000$$

where: ε = output strain in microinches/inch
V_1 = average excitation voltage
V_2 = Carlson-2 output voltage
R_{REF} = reference resistance ratio
Cal = calibration constant for the gage

The calibration constants and the reference resistance ratios are initialized at program startup to the values shown in table 1.

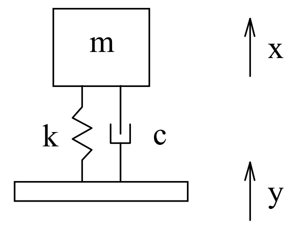
TABLE 1. CARLSON-2 STRAIN GAGE CALIBRATION CONSTANTS AND REFERENCE RESISTANCE RATIOS

Gage Number	Calibration	Ref. Ratio	Gage Number	Calibration	Ref. Ratio
A6511	3.40	0.9910	A6534	3.38	0.9911
A6512	3.39	0.9876	A6535	3.39	0.9946
A6513	3.39	0.9918	A6540	3.37	0.9923
A6520	3.40	0.9923	A6541	3.40	0.9937
A6521	3.38	0.9923	A6543	3.41	0.9926
A6522	3.43	0.9937	A6547	3.40	0.9932
A6523	3.39	0.9953	A6549	3.41	0.9931
A6524	3.38	0.9944	A6550	3.41	0.9951
A6529	3.39	0.9942	A6551	3.43	0.9942
A6530	3.41	0.9947	A6552	3.39	0.9883
A6531	3.39	0.9950	A6554	3.42	0.9883
A6532	3.41	0.9954			

2.18 GEOPHONE DATA PROCESSING.

The assumed mode of operation of the geophones is that a mass is suspended on a spring and a damper within a closed housing. The mass moves in the sensitive direction of the sensor and output from the sensor is a voltage proportional to the velocity of the mass relative to the housing.

$$\dot{z} = C \times V$$



where: $\dot{z} = \dot{x} - \dot{y}$ = relative velocity
C = calibration constant
V = output voltage
y = absolute displacement of the housing
x = absolute displacement of the mass

and the system equation of motion is

$$m\ddot{z} + c\dot{z} + kz = -m\ddot{y}, \text{ or}$$
$$\ddot{y} = -\ddot{z} - 2\xi\omega_n\dot{z} - \omega_n^2 z \tag{1}$$

where: m = mass of the proof mass
c = viscous damping coefficient
k = spring constant
ξ = damping ratio $\cong$ 0.7
ω_n = undamped natural frequency $\cong$ 20 rad/s (3 Hz)

See reference 1 or any text book on the theory of vibrations for more details.

The measured variable is $\dot{z}$ and equation 1 must be processed numerically to compute y as a function of time. The operations to be performed can be expressed as follows:

$$y = -\int \dot{z} - 2\xi\omega_n \iint \dot{z} - \omega_n^2 \iiint \dot{z} \tag{2}$$

where the integrations are all with respect to time.

Direct solution is possible by numerical integration, or a transfer function approach in the frequency domain can be used (reference 1 gives the transfer function). However, a direct understanding of the frequency response characteristics of numerically solving equation 2 can be obtained by solving the equation in the frequency domain term by term. Expressed in its Fourier components, $\dot{z}$ is represented by the expression

$$\dot{z} = \frac{a_0}{2} + \sum_{i=1}^{N/2} (a_i \cos\omega_i t + b_i \sin\omega_i t) \tag{3}$$

Also,

$$\int \cos\omega t.\mathrm{dt} = \frac{\sin\omega t}{\omega}; \quad \iint \cos\omega t.\mathrm{dt} = \frac{-\cos\omega t}{\omega^2}; \quad \iiint \cos\omega t.\mathrm{dt} = \frac{-\sin\omega t}{\omega^3}, \text{ and}$$
$$\int \sin\omega t.\mathrm{dt} = \frac{-\cos\omega t}{\omega}; \quad \iint \sin\omega t.\mathrm{dt} = \frac{-\sin\omega t}{\omega^2}; \quad \iiint \sin\omega t.\mathrm{dt} = \frac{\cos\omega t}{\omega^3}$$

Equation 3 can therefore be integrated term by term and the result substituted into equation 2 to compute housing displacement y. For one frequency and dropping the subscripts

$$y = \frac{1}{\omega}(b\cos\omega t - a\sin\omega t) + \frac{2\xi\omega_n}{\omega^2}(a\cos\omega t + b\sin\omega t) + \frac{\omega_n^2}{\omega^3}(-b\cos\omega t + a\sin\omega t) \qquad (4)$$

If the coefficients a and b in equation 3 are found using an FFT routine, the three terms in equation 4 can be found by simple interchange and sign change of the real and imaginary components of the FFT, and multiplication by the appropriate constants. (The negative frequency components are treated in the same way except that the sign changes are reversed.) Inverse transformation then gives y as a function of time. The following observations can also be made:

a. When ω is significantly greater than ω_n, the first term of equation 4 is dominant and is equivalent to a single integration of the measured signal. (Reference 1 refers to this as operation in the seismometer region.)

b. When ω is significantly less than ω_n, the third term of equation 4 is dominant and is equivalent to a triple integration of the measured signal. (Reference 1 refers to this as operation in the accelerometer region.)

c. When ω is close to ω_n, all three terms of equation 4 are of the same order.

Performing this procedure on geophone 3748V for B-727 run DEN003 (10 mph) gave the results in figures 5 through 9. The zero and lowest frequency Fourier components of the signal were set to zero after performing the FFT to provide some high-pass filtering. This amount of filtering gave the best shaped displacement signal. The amplitude of the component of y represented by each of the three terms increases significantly from the first term to the third. Figure 9 is the sum of the three terms and shows that the maximum change in y is almost completely determined by the third term in equation 4. The first two terms mainly serve to sharpen the signal and add a small phase shift.

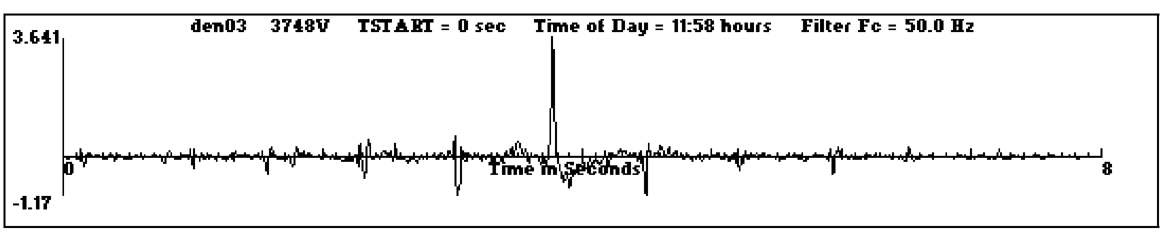


FIGURE 5. GEOPHONE SIGNAL AS RECORDED

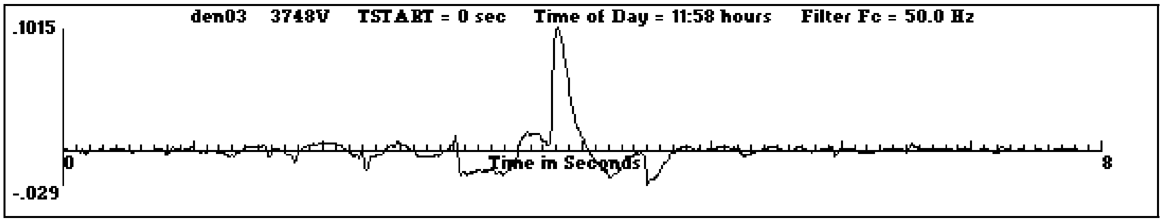


FIGURE 6. FIRST TERM OF EQUATION FOUR (FIRST INTEGRAL)

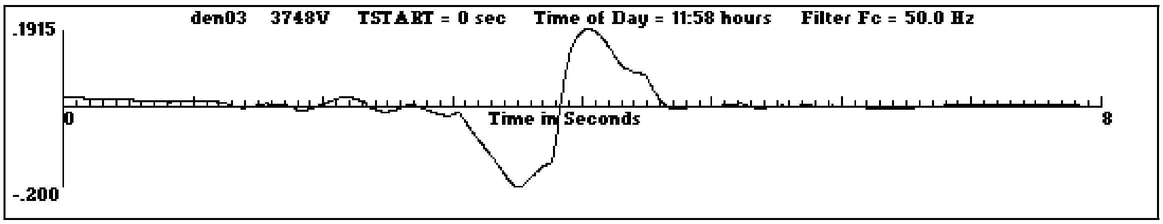


FIGURE 7. SECOND TERM OF EQUATION FOUR (SECOND INTEGRAL)

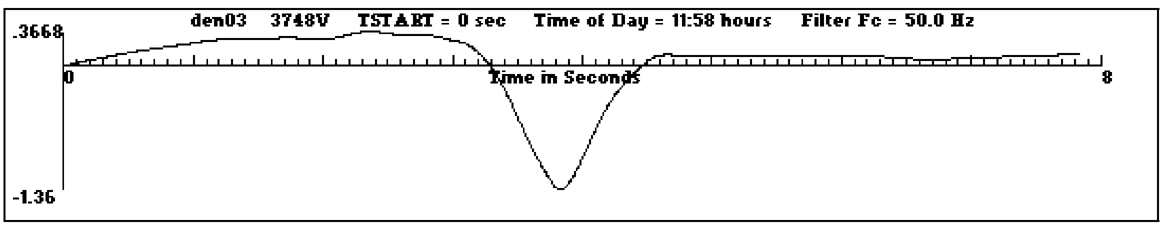


FIGURE 8. THIRD TERM OF EQUATION FOUR (THIRD INTEGRAL)

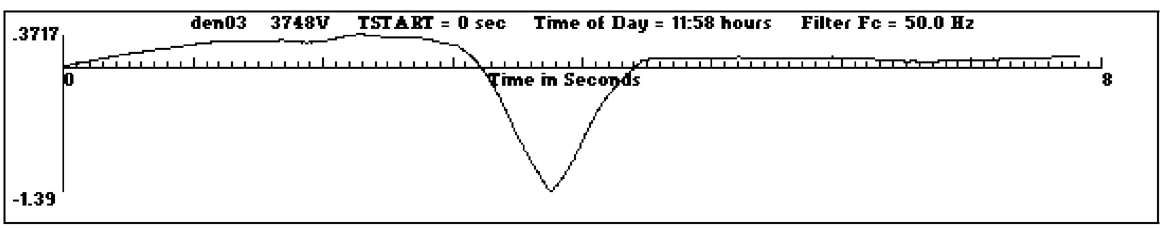


FIGURE 9. SUM OF THE SIGNALS IN FIGURES 3, 4, AND 5

For reference, figures 10 and 11 show the response of LVDT SDD14, which is located above the geophone, with figure 10 unfiltered and figure 11 low-pass filtered at 10 Hz. Figures 9 and 10 agree very well when the pavement is moving. But correspondence is, at least qualitatively, relatively poor outside the period when the pavement is moving.

These results show clearly that, for the characteristics of most interest, the geophone is operating well below the natural frequency of the instrument. Triple integration is therefore the dominant operation in the signal processing. Displacement measurement under these conditions will be very susceptible to low frequency errors such as those caused by hysteresis in the proof mass suspension.

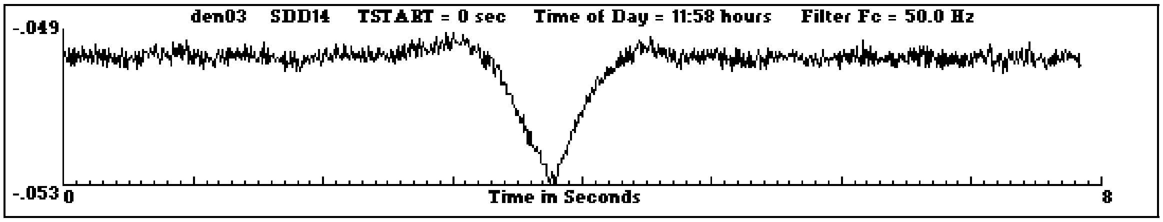


FIGURE 10. UNFILTERED LVDT SIGNAL

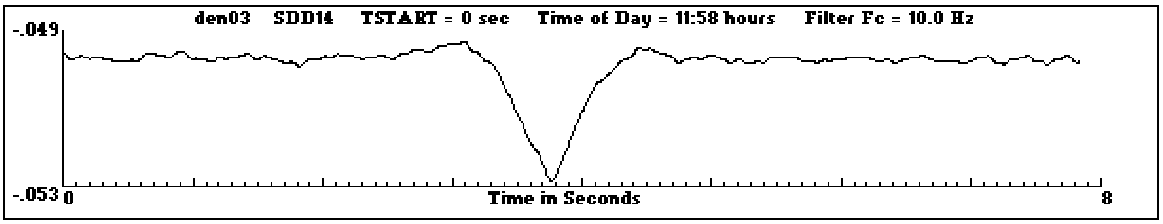


FIGURE 11. FILTERED LVDT SIGNAL

Geophone signal 3748V was chosen for the demonstration partly because it is located close to an LVDT and partly because it gives a nicely shaped signal in this particular test run. However, many of the other geophone signals show the effects of low-frequency errors, as shown in figures 12 and 13. It appears that the geophones will not be suitable for accurately measuring slab displacement unless a modification to the data processing can be found which will attenuate the low-frequency errors without excessively distorting the signal.

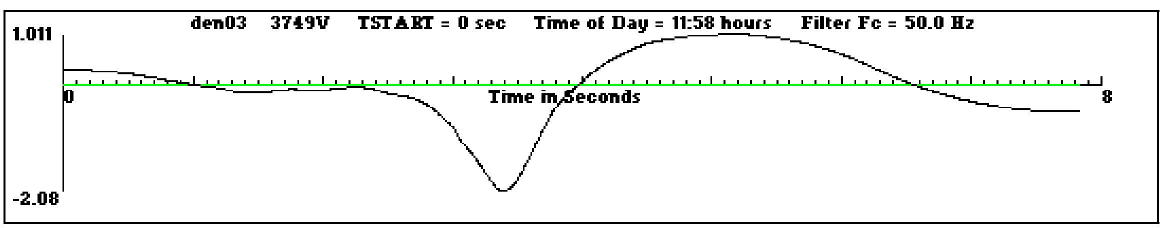


FIGURE 12. GEOPHONE 3749V

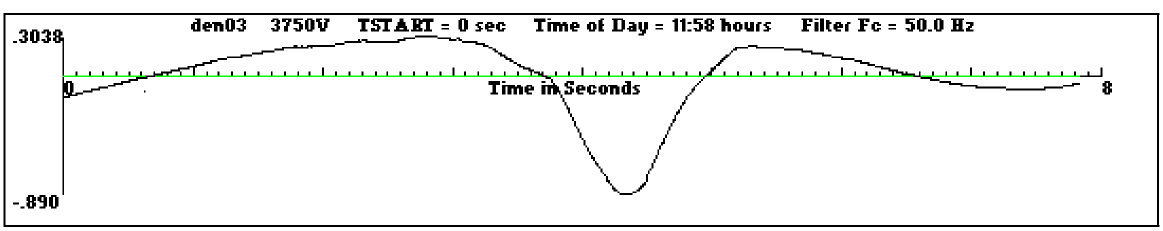


FIGURE 13. GEOPHONE 3750V

A further conclusion is that accelerometers would probably be more suitable instruments for measuring displacement without a deep anchor. The same signal processing procedure could be used, but only two integrations would be required, and frequency shaping would not be required.

3. STATIC SENSOR PROCESSING.

The static sensor data are stored in four separate groups, each within a separate file. The groups are:

a. Pavement temperature, resistivity, time domain reflectometer (TDR), Carlson strain, and Carlson joint data processed and stored by a Campbell data logger.

b. Weather station data.

c. Falling weight deflectometer (FWD) test data.

d. Aircraft traffic data.

All of the static sensor files are processed for loading into the database with a single Visual Basic program. The startup window is shown in figure 14. Clicking either of the Campbell, Weather, FWD, or Traffic command buttons displays a new window for processing the appropriate data file.

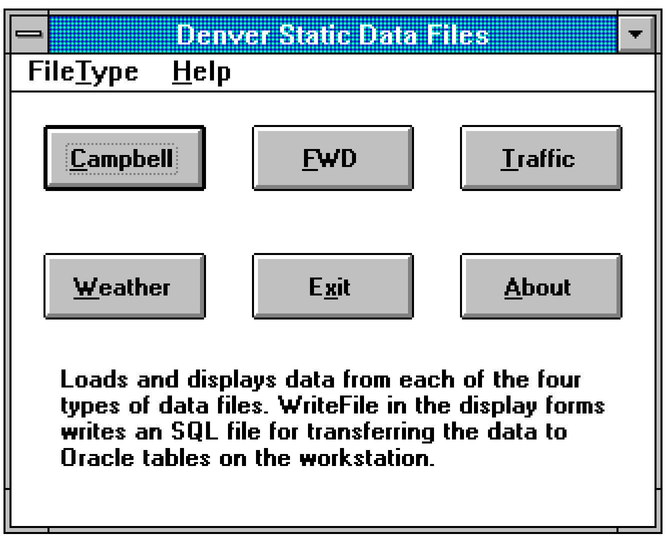


FIGURE 14. STATIC SENSOR SELECTION WINDOW

3.1 CAMPBELL DATA FILE.

Campbell file names have the format "DayTime.Cyr", where "DayTime" is Julian day and time in hours, minutes, and tenths of a minute, and "yr" is the last 2 digits of the year AD.

After the Campbell window has been displayed, a file can be selected from the File menu. All files with the extension .C?? in the currently selected path are shown in the file list box. After the file has been selected, it is opened, the data read and converted to the correct engineering units, and the converted data displayed in a text box.

Clicking WriteFile in the menu bar writes the data to an SQL file for transfer to the workstation and subsequent loading into the database. The same file name is used for the SQL file as the data file except that the extension is changed to .SQL.

3.1.1 Thermocouples and Resistivity Gages.

Thermoucouples and resistivity gages are arranged in three groups according to the concrete slab they are placed in or below. The data display for the slab B group is shown in figure 15. Thermocouple output is in degrees Fahrenheit and resistivity gage output is in ohms.

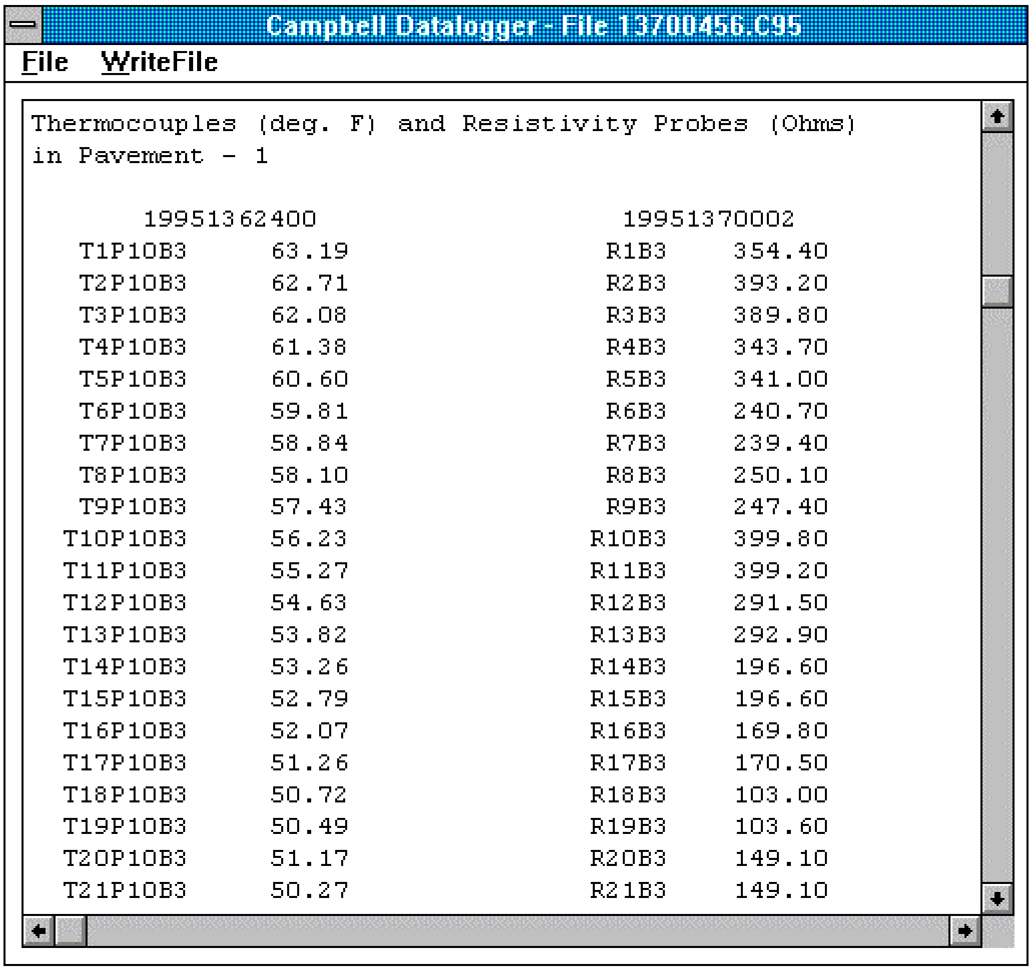


FIGURE 15. TYPICAL THERMOCOUPLE AND RESISTIVITY GAGE OUTPUT DISPLAY WINDOW

3.1.2 TDR Gage.

TDR gage raw data are stored in the Campbell file as the reciprocal of propagation velocity ($1/V_P$). Transformation to percent moisture content is by means of the Topps and Davis equation:

$$K_A = \left(\frac{1}{V_P} \right)^2$$

$$\text{Moisture} = -0.53 + 0.0292\ K_A - 0.00055\ K_A^2 + 0.0000043\ K_A^3$$

The data display is shown in figure 16.

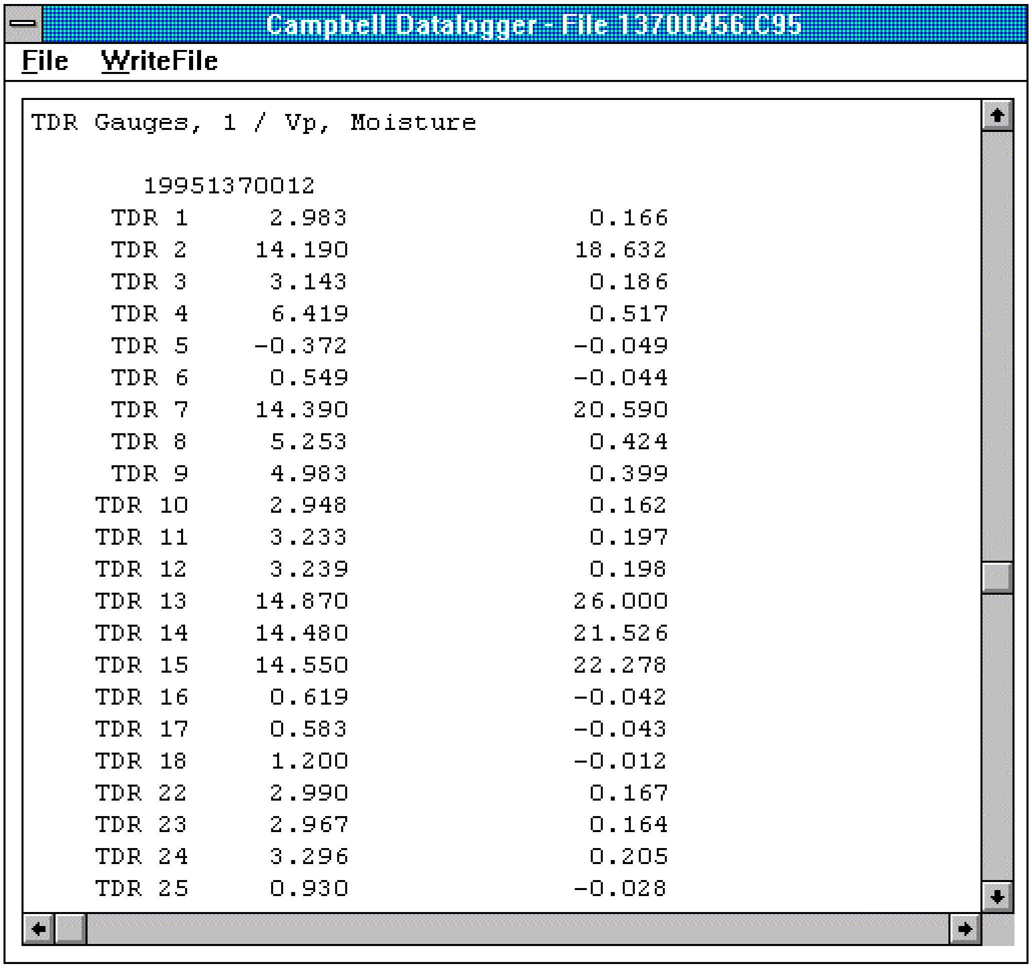


FIGURE 16. TYPICAL TDR GAGE DATA DISPLAY WINDOW

3.1.3 Carlson Strain and Joint Gage.

The Carlson strain and joint gage raw data are stored in the Campbell file as resistance ratios. Transformation to strain for the strain gages are

$$\varepsilon = \left(R - R_{REF} \right) \times Cal \times 10{,}000$$

where: ε = output strain in microinches/inch
R = measured resistance ratio
R_{REF} = reference resistance ratio
Cal = calibration constant for the gage

Transformation to displacement for the joint gages is

$$\delta = (R - R_{REF}) \times Cal \times 10{,}000{,}000$$

where: δ = output displacement in mils
R = measured resistance ratio
R_{REF} = reference resistance ratio
Cal = calibration constant for the gage

The calibration constants and the reference resistance ratios are initialized at program startup to the values in table 2.

TABLE 2. STATIC CARLSON STRAIN GAGE AND JOINT GAGE CALIBRATION CONSTANTS AND REFERENCE RESISTANCE RATIOS

Strain Gage	Calibration	Ref. Ratio	Joint Gage	Calibration	Ref. Ratio
A6510_R	3.44	0.9921	J897_R	0.00064	0.9940
A6514_R	3.39	0.9926	J898_R	0.00065	0.9906
A6515_R	3.40	0.9933	J899_R	0.00064	0.9973
A6516_R	3.40	0.9934	J900_R	0.00065	0.9934
A6517_R	3.38	0.9935	J901_R	0.00065	0.9933
A6518_R	3.41	0.9905	J902_R	0.00063	1.0073
A6519_R	3.44	0.9906	J903_R	0.00063	0.9992
A6525_R	3.43	0.9938	J904_R	0.00063	0.9987
A6526_R	3.43	0.9952	J905_R	0.00065	0.9983
A6527_R	3.42	0.9939	J906_R	0.00065	0.9966
A6528_R	3.39	0.9962			
A6533_R	3.39	0.9955			
A6536_R	3.39	0.9921			
A6537_R	3.39	0.9901			
A6542_R	3.38	0.9952			
A6545_R	3.37	0.9912			
A6546_R	3.39	0.9949			
A6548_R	3.40	0.9931			
A6553_R	3.44	0.9931			
A6555_R	3.40	0.9926			
A6556_R	3.39	0.9930			
A6563_R	3.44	0.9930			

Sensors which are not operating correctly are marked in the data logger file as having a resistance ratio of -1. This value is converted to strain or displacement values of 9999.99 in the display and the SQL file.

The data display is shown in figure 17.

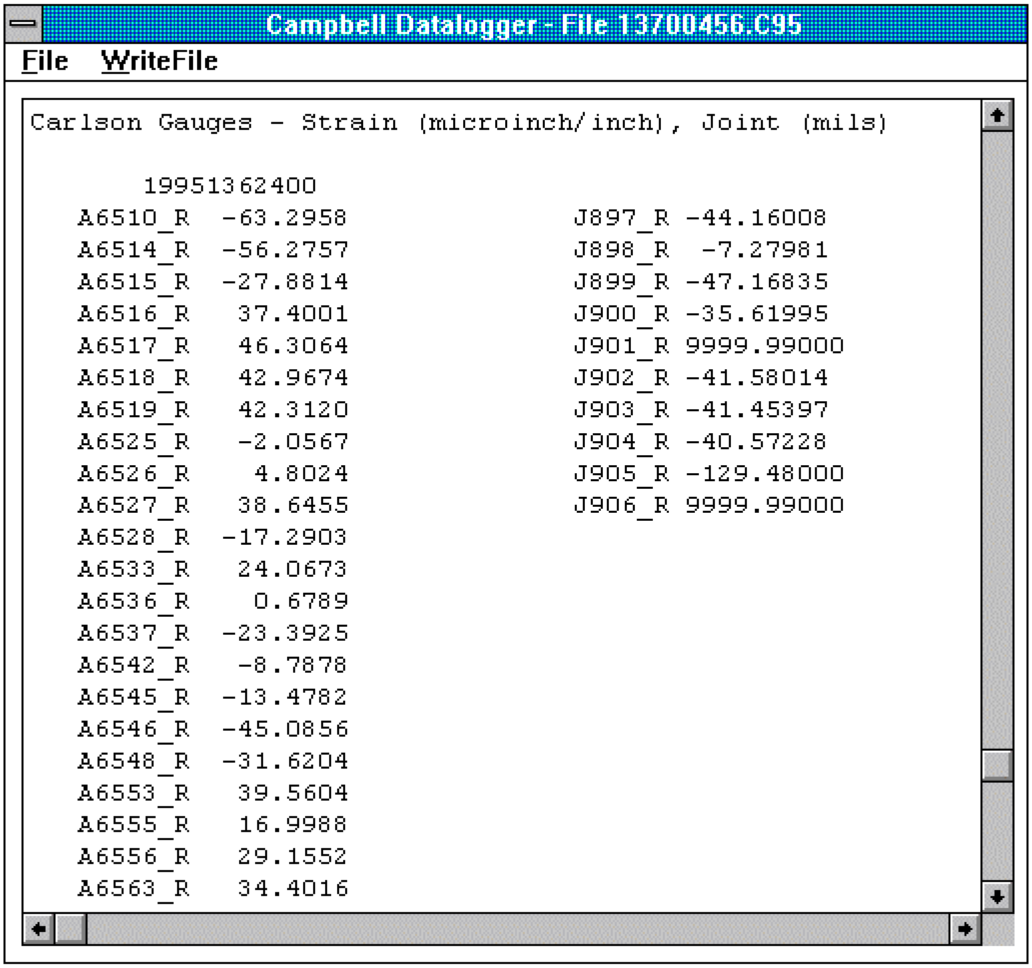


FIGURE 17. TYPICAL STATIC CARLSON GAGE OUTPUT DISPLAY WINDOW

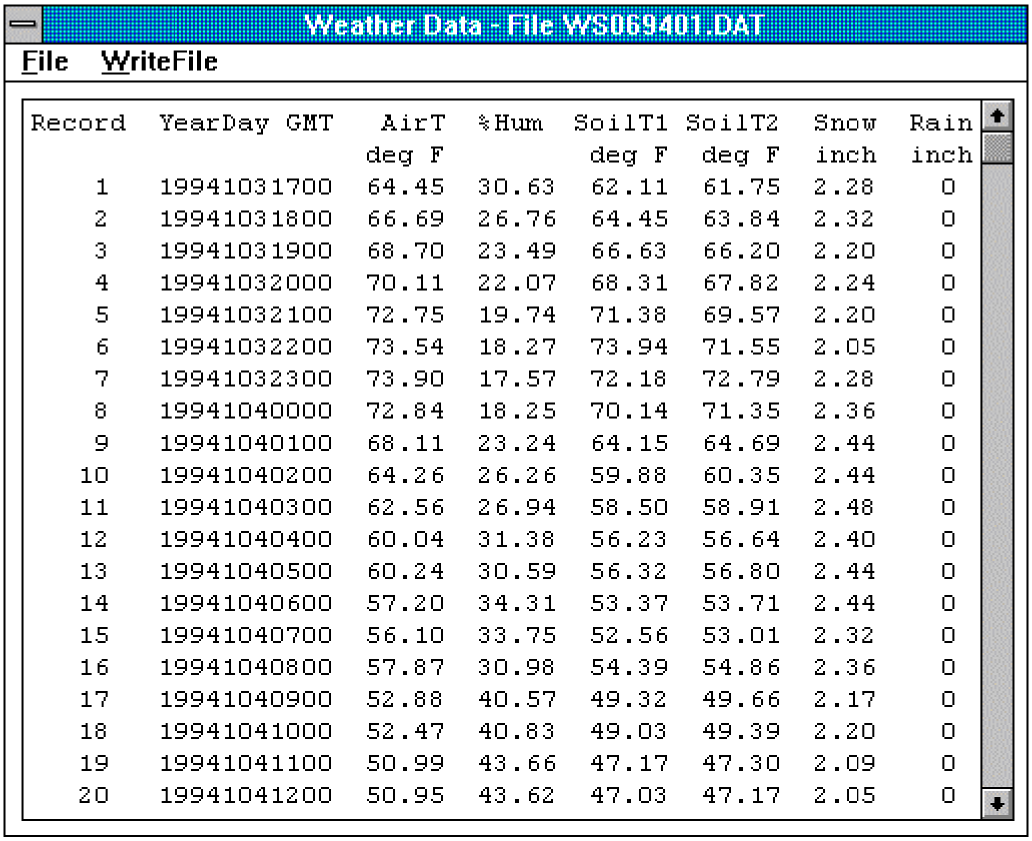
3.2 WEATHER STATION DATA FILE.

Weather station file names have the format "WS*.DAT."

After the Weather window has been displayed, a file can be selected from the File menu. All files with "WS" as the first two characters and extension .DAT in the currently selected path are shown in the file list box. After the file has been selected, it is opened, the data read and converted to the correct engineering units, and the converted data are displayed in a text box.

Clicking WriteFile in the menu bar writes the data to an SQL file for transfer to the workstation and subsequent loading into the database. The same file name is used for the SQL file as the data file except that the extension is changed to .SQL.

The data display and units after conversion are shown in figure 18. The column “Snow” is the depth of snow on the ground computed from a measurement of the distance from a set height above the ground to the surface of the snow. The set height is 1.245 metres (49.02 inches).


Weather Data - File WS069401.DAT

File WriteFile

Record	YearDay GMT	AirT deg F	%Hum	SoilT1 deg F	SoilT2 deg F	Snow inch	Rain inch
1	19941031700	64.45	30.63	62.11	61.75	2.28	0
2	19941031800	66.69	26.76	64.45	63.84	2.32	0
3	19941031900	68.70	23.49	66.63	66.20	2.20	0
4	19941032000	70.11	22.07	68.31	67.82	2.24	0
5	19941032100	72.75	19.74	71.38	69.57	2.20	0
6	19941032200	73.54	18.27	73.94	71.55	2.05	0
7	19941032300	73.90	17.57	72.18	72.79	2.28	0
8	19941040000	72.84	18.25	70.14	71.35	2.36	0
9	19941040100	68.11	23.24	64.15	64.69	2.44	0
10	19941040200	64.26	26.26	59.88	60.35	2.44	0
11	19941040300	62.56	26.94	58.50	58.91	2.48	0
12	19941040400	60.04	31.38	56.23	56.64	2.40	0
13	19941040500	60.24	30.59	56.32	56.80	2.44	0
14	19941040600	57.20	34.31	53.37	53.71	2.44	0
15	19941040700	56.10	33.75	52.56	53.01	2.32	0
16	19941040800	57.87	30.98	54.39	54.86	2.36	0
17	19941040900	52.88	40.57	49.32	49.66	2.17	0
18	19941041000	52.47	40.83	49.03	49.39	2.20	0
19	19941041100	50.99	43.66	47.17	47.30	2.09	0
20	19941041200	50.95	43.62	47.03	47.17	2.05	0

FIGURE 18. TYPICAL WEATHER DATA FILE OUTPUT WINDOW

3.3 FWD DATA FILE.

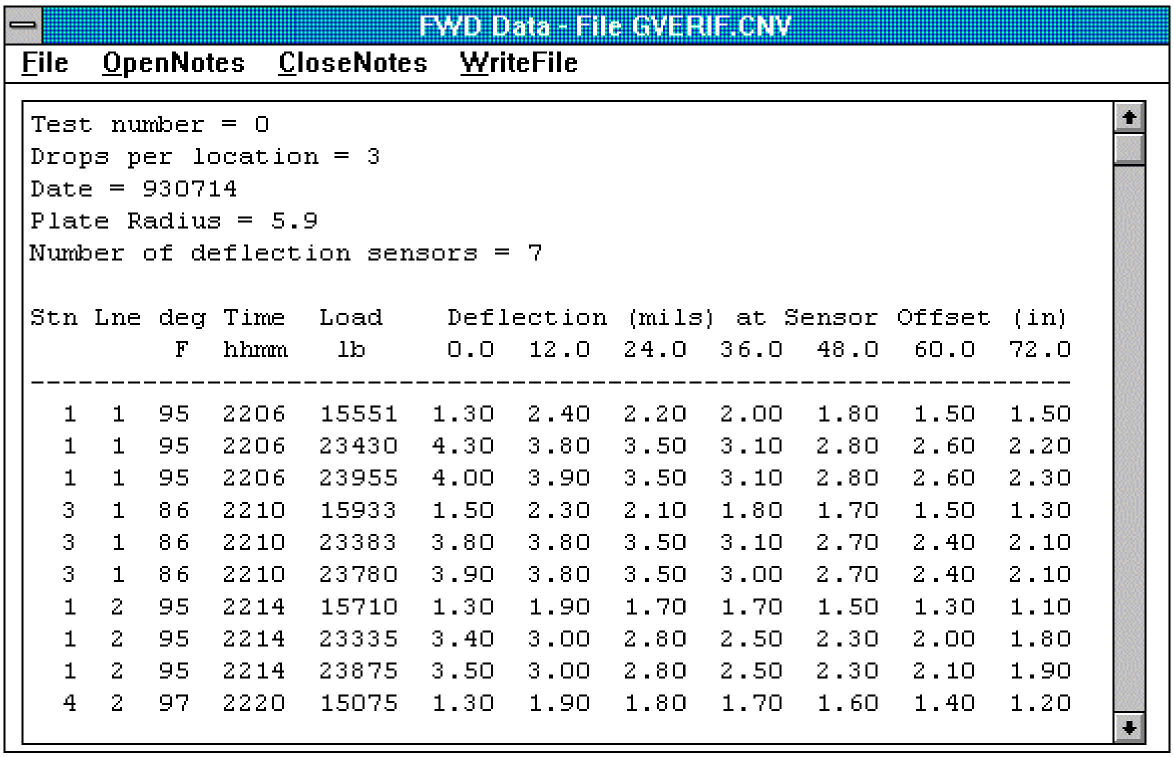
FWD file names have the extension .CNV.

After the FWD window has been selected and displayed, a file can be selected from the File menu. All files with “CNV” as the extension in the currently selected path are shown in the file list box. After the file has been selected, it is opened, the data read and converted to the correct engineering units, and the converted data are displayed in a text box.

Clicking OpenNotes in the Menu bar opens a second text box. Notes and comments can be entered in the text box. Clicking CloseNotes in the Menu bar closes the Notes text box and stores the information in the text box as a string. After transfer to the database, the Notes string is stored in the "note" column of the FWD_parameter table.

Clicking WriteFile in the menu bar writes the data to an SQL file for transfer to the workstation and subsequent loading into the database. The same file name is used for the SQL file as the data file except that the extension is changed to .SQL.

The data display and units after conversion are shown in figure 19.



```
FWD Data - File GVERIF.CNV
File   OpenNotes   CloseNotes   WriteFile

Test number = 0
Drops per location = 3
Date = 930714
Plate Radius = 5.9
Number of deflection sensors = 7

Stn Lne deg Time   Load    Deflection (mils) at Sensor Offset (in)
          F hhmm    lb     0.0   12.0  24.0  36.0  48.0  60.0  72.0
-----------------------------------------------------------------
  1   1  95  2206  15551  1.30  2.40  2.20  2.00  1.80  1.50  1.50
  1   1  95  2206  23430  4.30  3.80  3.50  3.10  2.80  2.60  2.20
  1   1  95  2206  23955  4.00  3.90  3.50  3.10  2.80  2.60  2.30
  3   1  86  2210  15933  1.50  2.30  2.10  1.80  1.70  1.50  1.30
  3   1  86  2210  23383  3.80  3.80  3.50  3.10  2.70  2.40  2.10
  3   1  86  2210  23780  3.90  3.80  3.50  3.00  2.70  2.40  2.10
  1   2  95  2214  15710  1.30  1.90  1.70  1.70  1.50  1.30  1.10
  1   2  95  2214  23335  3.40  3.00  2.80  2.50  2.30  2.00  1.80
  1   2  95  2214  23875  3.50  3.00  2.80  2.50  2.30  2.10  1.90
  4   2  97  2220  15075  1.30  1.90  1.80  1.70  1.60  1.40  1.20
```

FIGURE 19. TYPICAL FWD DATA FILE OUTPUT WINDOW

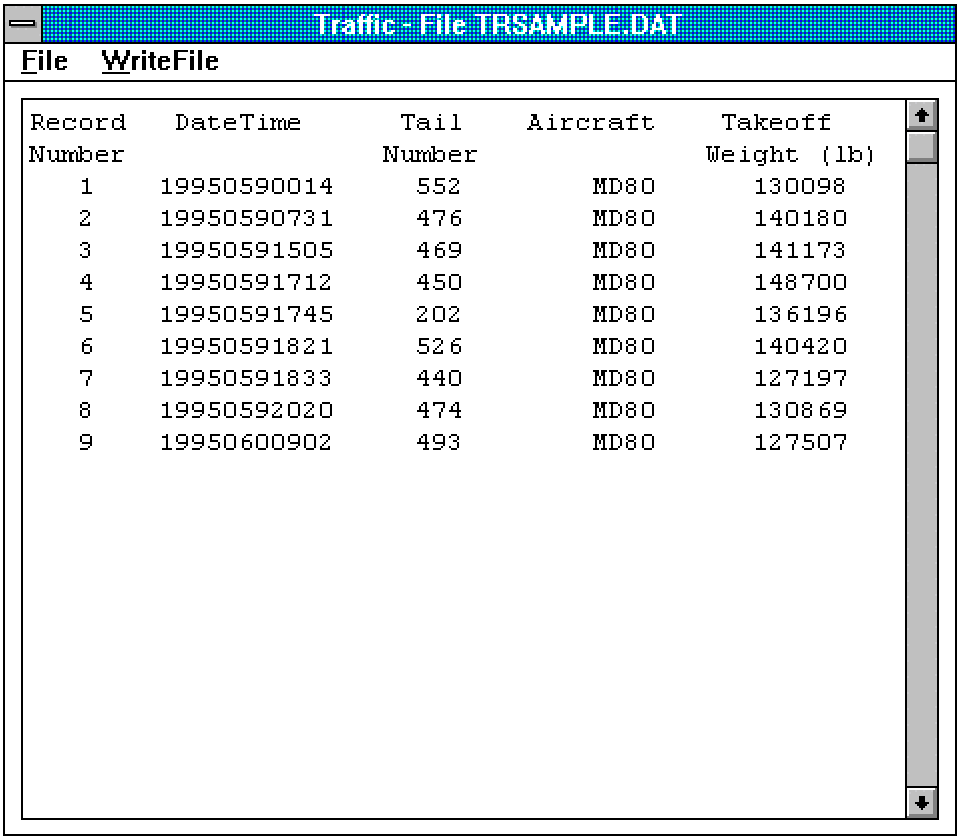
3.4 TRAFFIC DATA FILE.

Aircraft traffic file names have the format TR*.DAT.

After the FWD window has been selected and displayed, a file can be selected from the File menu. All files with "TR" as the first two characters and extension .DAT in the currently selected path are shown in the file list box. After the file has been selected, it is opened, the data read and converted to the correct engineering units, and the converted data are displayed in a text box.

Clicking WriteFile in the menu bar writes the data to an SQL file for transfer to the workstation and subsequent loading into the database. The same file name is used for the SQL file as the data file except that the extension is changed to .SQL.

Note: The format of the aircraft traffic files had not been finalized at the time this report was written. The data display and units for a sample file which contains data likely to be stored in the traffic files after the format has been finalized are shown in figure 20.



Record Number	DateTime	Tail Number	Aircraft	Takeoff Weight (lb)
1	19950590014	552	MD80	130098
2	19950590731	476	MD80	140180
3	19950591505	469	MD80	141173
4	19950591712	450	MD80	148700
5	19950591745	202	MD80	136196
6	19950591821	526	MD80	140420
7	19950591833	440	MD80	127197
8	19950592020	474	MD80	130869
9	19950600902	493	MD80	127507

FIGURE 20. TYPICAL TRAFFIC DATA FILE OUTPUT WINDOW

4. PERSONAL COMPUTER TO WORKSTATION COMMUNICATION.

The personal computer (PC) and the workstation communicate via a Netware LAN and VisionWare XVision installed on the PC. The general procedure for loading the database is to transfer the SQL files directly to the workstation from the PC and then execute the SQL files. However, operation of the database is also possible from the PC by running SQL-DBA, SQL*PLUS, or SQL Server Manager through the XVision X-Windows application.

4.1 TRANSFERRING FILES.

From the Windows Program Manager on the PC, double-click the **VisionWare** icon to display the **VisionWare** group.

Double-click the **File Transfer** icon to display the **File Transfer** window (see below). Then:

a. Open the Host list in the **From** frame by clicking the down arrow and select **local**.

b. Open the Host list in the **To** frame by clicking the down arrow and select **IRIS**.

c. Click the **Expand** command button to connect the PC and the workstation.

d. Type the specification of the file to be transferred in the **From** frame Filename box, or select the file name and path from the Files and Directories list boxes.

e. Select the directory to which the file will be transferred on the workstation in the **To** frame Directories box.

f. If the same file name is to be used on the workstation, type *.* in the **To** Filename box after the directory. Otherwise, type in a new file name.

g. Select Text as the File type in the **Options** frame.

h. Click the Transfer command button to transfer the file.

4.2 LOADING THE SQL FILE DATA INTO THE DATABASE.

The data in an SQL file is loaded into the database tables by executing the SQL file from within SQL*PLUS or Server Manager. When done on the workstation, a Unix shell must be opened after starting the computer. This is done by:

a. Click **Desktop** from the **Tools toolchest** at the top left of the screen. A menu will show on the screen.

b. Click **Unix Shell** on the menu. A Unix shell window will be loaded and displayed. The command prompt on the window is iris n%, where n is the command sequence number.

4.2.1 Loading From SQL*PLUS.

a. Connect to SQL*PLUS by typing **sqlplus** followed by Enter at the Unix shell prompt. Enter your Oracle user name and password when prompted. If SQL*PLUS connects successfully, the command prompt will change to SQL>.

b. Execute the SQL file with the **start** command by typing **start** followed by the path and filename of the SQL file. For example:

   start /usr/oracle/dong/sqlfile/43481166.sql

c. The new data loaded into the database can be checked with the **select** command. For example:

   select * from dong.aircraft where datetime = '19943481166';

d. Repeat a. and b. for all SQL files to be loaded.

e. Quit SQL*PLUS by typing **exit**.

4.2.2 Loading From Server Manager.

a. Connect to Server Manager by typing **svrmgrm** followed by Enter at the Unix shell prompt. The Server Manager startup window will show on the screen. Close the startup window by clicking it.

b. A dialog box will be displayed for connecting to the database. Enter your Oracle user name and password in the text boxes and click the **Connect** command button. If SQL*PLUS connects successfully, the Server Administrator window will be displayed.

c. From the **File** menu, select **Worksheet**. A worksheet window will be displayed.

d. Enter the same SQL commands to load and check the data as in SQL*PLUS. The commands are executed by clicking the **Execute** command button.

e. Quit Server Manager by selecting **Quit** from the Server Administrator **File** menu.

4.3 RETRIEVING SIGNAL PEAK RECORDS.

Time histories of selected peak records are saved in the event_records table. When needed, the peak records can be retrieved for analysis or to be plotted from a spread sheet or other document. The records are saved in the database as a text string in a LONG variable. Numerical values are saved sequentially with comma delimiters. The format is number of samples, time step length, sample 1, sample 2, etc. There are many different methods which can be used to retrieve and plot the records. The following is a straightforward procedure for transferring the data to an Excel spreadsheet.

a. Transfer the data to a file on the workstation.

This is done by using the spool and some other Oracle Report features. A few SQLPlus and SQL statements are needed to set up the report layout and to extract the data from the table. An example file for this purpose is listed as follows:

```
=============================
set long 2000
set longchunk 2000
set page 30
spool out8
select eventrecord from event_records
       where event#=61
       and sensor#='MDD9G1';
spool off
=============================
```

The reason for these settings is because a signal record is saved in the database in a LONG type variable and is composed of a few hundred ASCII characters. The default size of a LONG variable is 80, which allows only part of the signal to be displayed on the screen or transferred to a file. By setting the LONG as 2000, it will allow a 2000 character variable to be displayed. The LONGCHUNK and PAGE settings make the next steps easier.

To run this file, type start *filename* followed by return in the SQLPLus environment.

The name of the output file is out8, as specified in the SQL file above, and an extension .lst is added automatically. A listing of the file out8.lst is as follows:

```
=============================
1* select eventrecord from event_records where event#=61 and sensor#='MDD9G1'
SQL> /

EVENTRECORD
--------------------------------------------------------------------------------
209,0.019253,0.028,-0.022,0.024,0.152,0.227,0.192,0.128,0.113,0.110,0.055,-0.013
,0.014,0.134,0.192,0.085,-0.071,-0.103,-0.009,0.062,0.018,-0.103,-0.190,-0.159,-
0.045,0.028,0.031,0.068,0.138,0.063,-0.220,-0.472,-0.436,-0.196,-0.074,-0.245,-0
.576,-0.794,-0.785,-0.700,-0.728,-0.834,-0.841,-0.724,-0.636,-0.623,-0.519,-0.26
2,-0.090,-0.215,-0.485,-0.556,-0.303,0.064,0.256,0.216,0.120,0.104,0.082,-0.062,
-0.214,-0.153,0.100,0.255,0.145,-0.010,0.089,0.415,0.679,0.703,0.591,0.538,0.572
,0.549,0.394,0.252,0.283,0.396,0.412,0.359,0.375,0.428,0.359,0.143,-0.077,-0.232
,-0.369,-0.467,-0.460,-0.436,-0.542,-0.775,-0.983,-1.063,-1.083,-1.175,-1.382,-1
.649,-1.885,-2.038,-2.240,-2.856,-4.176,-5.926,-7.301,-7.701,-7.271,-6.642,-6.27
7,-6.112,-5.820,-5.306,-4.806,-4.528,-4.388,-4.188,-3.858,-3.458,-3.063,-2.700,-
2.359,-2.050,-1.797,-1.566,-1.311,-1.089,-0.973,-0.868,-0.621,-0.257,0.047,0.213
,0.342,0.515,0.625,0.506,0.220,0.047,0.142,0.346,0.421,0.336,0.218,0.130,0.063,0
.041,0.066,0.061,0.008,0.001,0.074,0.121,0.080,0.042,0.071,0.082,0.005,-0.074,-0
.061,-0.017,-0.059,-0.177,-0.253,-0.205,-0.077,0.023,0.057,0.072,0.075,0.033,-0.
024,-0.041,-0.037,-0.057,-0.069,-0.010,0.093,0.110,-0.023,-0.191,-0.217,-0.054,0
.162,0.251,0.164,0.002,-0.124,-0.151,-0.038,0.170,0.267,0.072,-0.270,-0.410,-0.2
53,-0.067,-0.079,-0.204,-0.259,-0.207,-0.116,-0.031,0.026,0.012,-0.104,-0.263,-0
.323,-0.204,-0.023,0.040

 SQL> spool off
=============================
```

In the above file, only the numbers are required to plot the curve. The first two numbers represent the number of data points (209 in this example) and the time axis increment (0.019253 in this example), respectively. The rest of the numbers are sensor data values (dependent axis).

b. Plot the curve using Microsoft Excel

(1) The file (out8.lst in this example) is loaded into a word processor, for example, Microsoft Word.

(2) Unnecessary characters are deleted so that numerals and commas only are left.

(3) Highlight the numerals and commas.

(4) Convert the text to a table by clicking the TABLE and CONVERT TEXT TO TABLE menu items.

(5) Copy the converted table into Microsoft Excel.

(6) Add the incremented data of the horizontal axis.

(7) Set up the layout and plot the curve, as illustrated in figure 21.

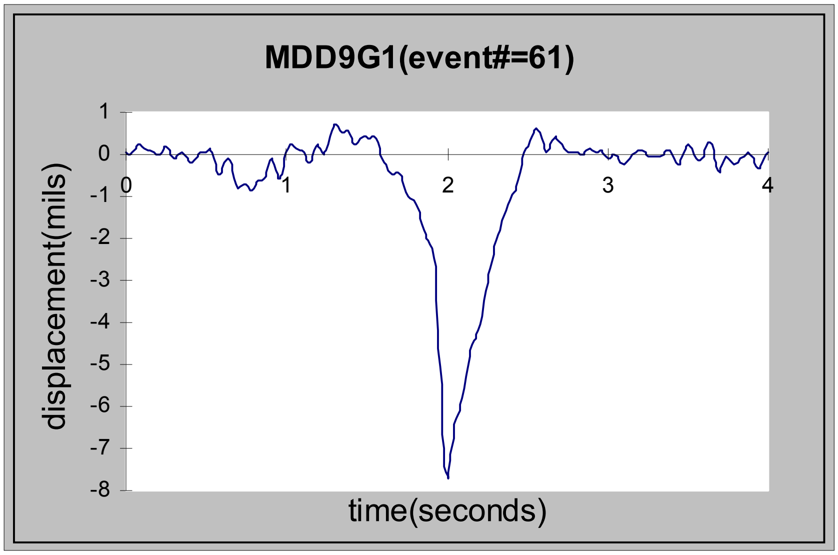


FIGURE 21. PEAK RECORD PLOTTED USING THE DATA FROM THE ORACLE DATABASE

5. DATABASE STRUCTURE.

5.1 INTRODUCTION.

All of the information for the dynamic and static sensor measurements is stored in a single database. Other measurements of pavement properties and condition, not previously described, are also stored in the database. Various links are provided so that the different types of information can be referenced to each other. However, because of the diversity of the information, date is the only common reference across all tables in the database. In the following description of the database structure, the database tables are separated into five groups:

a. Dynamic Sensor Response
b. Static Sensor Response
c. Dynamic and Static Sensor Information
d. Miscellaneous
e. Falling Weight Deflectometer (FWD) Tests

A brief description is first given for each group. This is followed by tabular listings of the structure of each database table. Where there are multiple tables of the same structure for a single type of sensor, only the typical structure is given. Column names and data formats are listed with measurement units and comments where necessary.

Table names are denoted by **bold** type and column names by *italic* type.

5.1.1 Group 1, Dynamic Sensor Response—Summary.

The primary table in Group 1 is the **aircraft** table and the primary key in the **aircraft** table is *event#*. *Event#* is incremented automatically each time an aircraft event is stored in the database. However, if an event has been deleted from the database, for any reason, the higher event numbers are not adjusted to maintain consecutive numbering and gaps may therefore occur. Renumbering is not done in order to maintain the same identification for a given physical event over time. An aircraft event is defined as the passage of one aircraft across the instrumented pavement and includes all sensor data values, occurrence times, and aircraft data for the pass stored in the database. The **aircraft** table has a *note* column for entering incidental information about any given event.

Peak response data for each dynamic sensor is stored in a separate table having the same name as the sensor. Reference to the **aircraft** table is by *event#*. Each dynamic sensor table has a note column for entering incidental information about the sensor, such as changes in calibration. An additional table, **event_records,** is used to store the peak record time histories whenever they have been saved during data processing.

5.1.1.1 Table Structures—Group 1, Dynamic Sensor Response.

Table aircraft

event# number(38) primary key		
datetime char(11) YYYYDDDhhmm		
seconds varchar2(2)		
aircraft# varchar2(20) not null		
aircraftweight number(7)	lb	
carrier varchar2(20)	lb	
NvalidIR number(1)		
v0 number(6,3)	ft/s	
accel number(7,3)	ft/s^2	
yposition number(7,3)	ft	
note varchar2(1000)		
acpicture varchar2(50) path & filename		
filename varchar2(20)		
aircrafttype varchar2(20)		

Table weather

datetime	char(11) primary key YYYYDDDhhmm	
airtemp	number(5,2)	degrees F
humidity	number(5,2)	percent
soiltem1	number(5,2)	degrees F
soiltem2	number(5,2)	degrees F
snowfall	number(5.2)	inch
rainfall	number(5,2)	inch

Table event_records

event#	number(38) references aircraft(event#) on delete cascade	
sensor#	varchar2(12) not null	
eventrecord	long	

Table HB1

(one of H-Bar strain tables)

event#	number(38) references aircraft(event#) on delete cascade	
Tinterval	number(7,6)	sec/sample
Tstart	number(7,6))	second
Tend	number(5,3)	second
FilterFreq	number(5,3)	Hz
Npeaks	number(1)	
offsetLeft	number(10,3)	μ in/in
offsetRight	number(10,3)	μ in/in
time1	number(5,3) not null	second
peak1	number(7,3) not null	μ in/in
time2	number(5,3)	second
peak2	number(7,3)	μ in/in
time3	number(5,3)	second
peak3	number(7,3)	μ in/in
time4	number(5,3)	second
peak4	number(7,3)	μ in/in
note	varchar2(1000)	

Every dynamic sensor has one table for its values of peak response, the time at which the peak occurred, etc., such as table HB1. Each event occupies one row in these tables. The sum total of dynamic sensor tables is 225 including 36 position tables, 102 H-Bar tables, 50 LVDT tables, 23 Carlson tables and 14 Geophone tables. Peak record data is stored in the event_records table. The format of the column eventrecord is an ASCII string containing 'the number of samples, sample interval, value1, value2,..., lastvalue'. Consecutive items are separated by spaces.

Table P1
(one of position strain tables)

event# number(38) references aircraft(event#) on delete cascade		
Tinterval	number(7,6)	sec/sample
Tstart	number(7,6))	second
Tend	number(5,3)	second
FilterFreq	number(5,3)	Hz
Npeaks	number(1)	
offsetLeft	number(10,3)	µ in/in
offsetRight	number(10,3)	µ in/in
time1	number(5,3) not null	second
peak1	number(7,3) not null	µ in/in
time2	number(5,3)	second
peak2	number(7,3)	µ in/in
time3	number(5,3)	second
peak3	number(7,3)	µ in/in
time4	number(5,3)	second
peak4	number(7,3)	µ in/in
note	varchar2(1000)	

Table MDD1G1
(one of SDD and MDD LVDT tables)

event# number(38) references aircraft(event#) on delete cascade		
Tinterval	number(7,6)	sec/sample
Tstart	number(7,6))	second
Tend	number(5,3)	second
FilterFreq	number(5,3)	Hz
Npeaks	number(1)	
offsetLeft	number(10,3)	mils
offsetRight	number(10,3)	mils
time1	number(5,3) not null	second
peak1	number(7,3) not null	mils
time2	number(5,3)	second
peak2	number(7,3)	mils
time3	number(5,3)	second
peak3	number(7,3)	mils
time4	number(5,3)	second
peak4	number(7,3)	mils
note	varchar2(1000)	

Table G3738H
(one of Geophone tables)

event# number(38) references aircraft(event#) on delete cascade		
Tinterval	number(7,6)	sec/sample
Tstart	number(7,6))	second
Tend	number(5,3)	second
FilterFreq	number(5,3)	Hz
Npeaks	number(1)	
offsetLeft	number(10,3)	volts
offsetRight	number(10,3)	volts
time1	number(5,3) not null	second
peak1	number(7,3) not null	volts
time2	number(5,3)	second
peak2	number(7,3)	volts
time3	number(5,3)	second
peak3	number(7,3)	volts
time4	number(5,3)	second
peak4	number(7,3)	volts
note	varchar2(1000)	

Table A6511
(one of dynamic Carlson strain tables)

event# number(38) references aircraft(event#) on delete cascade		
Tinterval	number(7,6)	sec/sample
Tstart	number(7,6))	second
Tend	number(5,3)	second
FilterFreq	number(5,3)	Hz
Npeaks	number(1)	
offsetLeft	number(10,3)	µ in/in
offsetRight	number(10,3)	µ in/in
time1	number(5,3) not null	second
peak1	number(7,3) not null	µ in/in
time2	number(5,3)	second
peak2	number(7,3)	µ in/in
time3	number(5,3)	second
peak3	number(7,3)	µ in/in
time4	number(5,3)	second
peak4	number(7,3)	µ in/in
note	varchar2(1000)	

5.1.2 Group 2, Static Sensor Response—Summary.

Campbell data logger data is stored in 10 tables, with each sensor having its data value stored in one column of one of the tables.

5.1.2.1 Table Structures—Group 2, Static Sensor Response.

Table Strain_AR

datetime	char(11) primary key YYYYDDDhhmm	
A6510_R	number(6,2)	µ in/in
A6514_R	number(6,2)	
A6515_R	number(6,2)	
A6516_R	number(6,2)	
A6517_R	number(6,2)	
A6518_R	number(6,2)	
A6519_R	number(6,2)	
A6525_R	number(6,2)	
A6526_R	number(6,2)	
A6527_R	number(6,2)	
A6528_R	number(6,2)	
A6533_R	number(6,2)	
A6536_R	number(6,2)	
A6537_R	number(6,2)	
A6542_R	number(6,2)	
A6545_R	number(6,2)	
A6546_R	number(6,2)	
A6548_R	number(6,2)	
A6553_R	number(6,2)	
A6555_R	number(6,2)	
A6556_R	number(6,2)	
A6563_R	number(6,2)	

Table Joint_JR

datetime	char(11) primary key YYYYDDDhhmm	
J897_R	number(7,2)	mils
J898_R	number(7,2)	
J899_R	number(7,2)	
J900_R	number(7,2)	
J901_R	number(7,2)	
J902_R	number(7,2)	
J903_R	number(7,2)	
J904_R	number(7,2)	
J905_R	number(7,2)	
J906_R	number(7,2)	

The total number of Campbell tables is 10, including 3 thermocouple tables, 3 resistivity tables, 2 TDR tables, 1 strain table, and 1 joint table. They relate to other tables by column datetime.

Table RGRND

datetime char(11) primary key YYYYDDDhhmm		
R1GRND	number(6,2)	ohms
R2GRND	number(6,2)	
R3GRND	number(6,2)	
R4GRND	number(6,2)	
R5GRND	number(6,2)	
R6GRND	number(6,2)	
R7GRND	number(6,2)	
R8GRND	number(6,2)	
R9GRND	number(6,2)	
R10GRND	number(6,2)	
R11GRND	number(6,2)	
R12GRND	number(6,2)	
R13GRND	number(6,2)	
R14GRND	number(6,2)	
R15GRND	number(6,2)	
R16GRND	number(6,2)	
R17GRND	number(6,2)	
R18GRND	number(6,2)	
R19GRND	number(6,2)	
R20GRND	number(6,2)	
R21GRND	number(6,2)	
R22GRND	number(6,2)	
R23GRND	number(6,2)	
R24GRND	number(6,2)	
R25GRND	number(6,2)	
R26GRND	number(6,2)	
R27GRND	number(6,2)	
R28GRND	number(6,2)	
R29GRND	number(6,2)	
R30GRND	number(6,2)	
R31GRND	number(6,2)	
R32GRND	number(6,2)	

Table TGRND

datetime char(11) primary key YYYYDDDhhmm		
T1GRND	number(5,2)	Degrees F
T2GRND	number(5,2)	
T3GRND	number(5,2)	
T4GRND	number(5,2)	
T5GRND	number(5,2)	
T6GRND	number(5,2)	
T7GRND	number(5,2)	
T8GRND	number(5,2)	
T9GRND	number(5,2)	
T10GRND	number(5,2)	
T11GRND	number(5,2)	
T12GRND	number(5,2)	
T13GRND	number(5,2)	
T14GRND	number(5,2)	
T15GRND	number(5,2)	
T16GRND	number(5,2)	
T17GRND	number(5,2)	
T18GRND	number(5,2)	
T19GRND	number(5,2)	
T20GRND	number(5,2)	

Table RB3

datetime char(11) primary key YYYYDDDhhmm		
R1B3	number(6,2)	Ohms
R2B3	number(6,2)	
R3N3	number(6,2)	
R4B3	number(6,2)	
R5B3	number(6,2)	
R6B3	number(6,2)	
R7B3	number(6,2)	
R8B3	number(6,2)	
R9B3	number(6,2)	
R10B3	number(6,2)	
R11B3	number(6,2)	
R12B3	number(6,2)	
R13B3	number(6,2)	
R14B3	number(6,2)	
R15B3	number(6,2)	
R16B3	number(6,2)	
R17B3	number(6,2)	
R18B3	number(6,2)	
R19B3	number(6,2)	
R20B3	number(6,2)	
R21B3	number(6,2)	
R22B3	number(6,2)	
R23B3	number(6,2)	
R24B3	number(6,2)	
R25B3	number(6,2)	
R26B3	number(6,2)	
R27B3	number(6,2)	
R28B3	number(6,2)	
R29B3	number(6,2)	
R30B3	number(6,2)	
R31B3	number(6,2)	
R32B3	number(6,2)	

Table TB3

datetime char(11) primary key YYYYDDDhhmm		
T1P10B3	number(5,2)	Degrees F
T2P10B3	number(5,2)	
T3P10B3	number(5,2)	
T4P10B3	number(5,2)	
T5P10B3	number(5,2)	
T6P10B3	number(5,2)	
T7P10B3	number(5,2)	
T8P10B3	number(5,2)	
T9P10B3	number(5,2)	
T10P10B3	number(5,2)	
T11P10B3	number(5,2)	
T12P10B3	number(5,2)	
T13P10B3	number(5,2)	
T14P10B3	number(5,2)	
T15P10B3	number(5,2)	
T16P10B3	number(5,2)	
T17P10B3	number(5,2)	
T18P10B3	number(5,2)	
T19P10B3	number(5,2)	
T20P10B3	number(5,2)	
T21P10B3	number(5,2)	
T22P10B3	number(5,2)	
T1P2B3	number(5,2)	
T2P2B3	number(5,2)	
T3P2B3	number(5,2)	
T4P2B3	number(5,2)	
T5P2B3	number(5,2)	
T6P2B3	number(5,2)	
T7P2B3	number(5,2)	
T8P2B3	number(5,2)	

Table RC3 has the same structure as **table RB3**.
Table TC3 has the same structure as **table TB3**.

Table TDR_OneByVp

datetime	char(11) primary key YYYYDDDhhmm	
TDR1	number(6,3)	
TDR2	number(6,3)	
TDR3	number(6,3)	
TDR4	number(6,3)	
TDR5	number(6,3)	
TDR6	number(6,3)	
TDR7	number(6,3)	
TDR8	number(6,3)	
TDR9	number(6,3)	
TDR10	number(6,3)	
TDR11	number(6,3)	
TDR12	number(6,3)	
TDR13	number(6,3)	
TDR14	number(6,3)	
TDR15	number(6,3)	
TDR16	number(6,3)	
TDR17	number(6,3)	
TDR18	number(6,3)	
TDR22	number(6,3)	
TDR23	number(6,3)	
TDR24	number(6,3)	
TDR25	number(6,3)	
TDR26	number(6,3)	
TDR27	number(6,3)	
TDR28	number(6,3)	
TDR29	number(6,3)	
TDR30	number(6,3)	
TDR31	number(6,3)	
TDR32	number(6,3)	
TDR33	number(6,3)	
TDR36	number(6,3)	
TDR37	number(6,3)	
TDR38	number(6,3)	
TDR39	number(6,3)	
TDR40	number(6,3)	
TDR41	number(6,3)	
TDR42	number(6,3)	
TDR43	number(6,3)	
TDR44	number(6,3)	
TDR45	number(6,3)	
TDR46	number(6,3)	
TDR47	number(6,3)	
TDR48	number(6,3)	
TDR49	number(6,3)	
TDR50	number(6,3)	

Table TDR_Moisture

datetime	char(11) primary key YYYYDDDhhmm	
TDR1	number(7,3)	percent
TDR2	number(7,3)	
TDR3	number(7,3)	
TDR4	number(7,3)	
TDR5	number(7,3)	
TDR6	number(7,3)	
TDR7	number(7,3)	
TDR8	number(7,3)	
TDR9	number(7,3)	
TDR10	number(7,3)	
TDR11	number(7,3)	
TDR12	number(7,3)	
TDR13	number(7,3)	
TDR14	number(7,3)	
TDR15	number(7,3)	
TDR16	number(7,3)	
TDR17	number(7,3)	
TDR18	number(7,3)	
TDR22	number(7,3)	
TDR23	number(7,3)	
TDR24	number(7,3)	
TDR25	number(7,3)	
TDR26	number(7,3)	
TDR27	number(7,3)	
TDR28	number(7,3)	
TDR29	number(7,3)	
TDR30	number(7,3)	
TDR31	number(7,3)	
TDR32	number(7,3)	
TDR33	number(7,3)	
TDR36	number(7,3)	
TDR37	number(7,3)	
TDR38	number(7,3)	
TDR39	number(7,3)	
TDR40	number(7,3)	
TDR41	number(7,3)	
TDR42	number7,3)	
TDR43	number(7,3)	
TDR44	number(7,3)	
TDR45	number(7,3)	
TDR46	number(7,3)	
TDR47	number(7,3)	
TDR48	number(7,3)	
TDR49	number(7,3)	
TDR50	number(7,3)	

5.1.3 Group 3, Dynamic and Static Sensor Information—Summary.

Unchanging information for the sensors is stored in "gage information tables." This information includes sensor manufacturer, position in the pavement, and any other information which may be of interest during analysis. The information varies with sensor type and each sensor type has a separate table.

Weather station data is stored in the **weather** table. The primary key is datetime.

5.1.3.1 Table Structures—Group 3, Dynamic and Static Sensor Information.

The sum total of gage information tables is nine including all sensor information. They relate to Campbell tables or dynamic sensor tables by column gage#.

Table Temperature_Gage

gage#	varchar2(8) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(4,2)	inch
probe_length	number(2)	inch
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Resistivity_Gage

gage#	varchar2(5) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(4,2)	inch
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table TDR_Probe

gage#	varchar2(5) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(4,2)	inch
orientation	char(3)	
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Strain_Gage

gage#	varchar2 (7) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(6,4)	inch
orientation	varchar2(4)	
calibration	number(3,2)	percent
ratio_reference	number(4,2)	
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Joint_Gage

gage#	varchar2(6) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(5,3)	inch
orientation	char(3)	
calibration	number(6,5)	percent
ratio_reference	number(5,2)	
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Hbar_Gage

gage#	varchar2(8) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(6,4)	inch
orientation	varchar2(8)	
calibration	number(5,2)	μ in/in per.volt
loc_code	char(4)	
cross_ref	varchar2(8)	
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Position_Gage

gage#	varchar2(3) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(2,1)	inch
orientation	char(8)	
calibration	number(5,2)	μ in/in/volt
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table LVDTs_Gage

gage#	varchar2(7) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(5,3)	inch
anchor_depth	number(5,2)	inch
calibration	number(5,3)	volts DC/in
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

Table Geophone_Gage

gage#	char(6) primary key	
slab#	char(2)	
slab_north	number(6,4)	feet
slab_east	number(6,4)	feet
denver_north	number(7,2)	feet
denver_east	number(7,2)	feet
pcc_thickness	number(4,2)	inch
depth	number(4,2)	inch
orientation	varchar2(10)	
calibration	number(3,2)	volts/in/sec
installation_date	date	
manufacturer	varchar2(8)	
note	varchar2(1000)	

5.1.4 Group 4, Miscellaneous—Summary.

Tables for generally unrelated information are listed here. Topics covered are aircraft traffic; pavement condition surveys; manual elevation measurements; neutron probes; laboratory tests; and backcalculated pavement properties.

5.1.4.1 Table Structures—Group 4, Miscellaneous.

These tables contain data that consists of traffic pavement condition index (PCI) survey, absolute surface elevation, water table height, and incidental laboratory measurements.

Table Traffic_Counter

datetime char(11) primary key	tail_number number(5)	equipment varchar2(10)	Takeoff_weightnumber(7)
YYYYDDDHHMM			Lb

Pavement Condition Survey Tables

Table survey_section_summary

survey_date	char(11) DDMMMYYYY	
section_size	number(6)	
random_sample_units	number(3)	
additional_sample_units	number(2)	
confidence	number(4,1)	percent
allowable_error	number(3,1)	
section_pci	number(4,1)	
standard_deviation	number(3,1)	
deduct_load	number(4,2)	
deduct_climate_durability	number(4,2)	percent
deduct_other	number(4,2)	
location	varchar2(50)	percent
note	varchar2(100)	

Table survey_unit_pci

survey_date	sampling_unit_id	pci	unit_size
char(11) DDMMMYYYY	number(3)	number(4,1)	number(3)

Table sample_unit1

survey_date	char(11) DDMMMYYYY	
distress_type	varchar2(20)	
severity	varchar2(10)	
quantiry	number(5,2)	
density	number(5,2)	percent
deduct_value	number(5,2)	

Table survey_section_distress

survey_date	char(11) DDMMMYYYY	
distress_type	varchar2(20)	
severity	varchar2(10)	
quantiry	number(5,2)	
density	number(5,2)	percent
deduct_value	number(5,2)	

Tables sample_unit2 through **sample_unit2** have the same structure as **table sample_unit1**.

Manual Elevation Measurement Tables

Table benchmark_and_wells

description	varchar2(20) not null	
denver_north	varchar2(15)	feet
denver_east	varchar2(15)	feet
elevation	varchar2(14)	feet
note	varchar2(18)	

Table brass_pin_data

survey_date	date	
time	varchar2(6)	feet
benchmark	varchar2(14)	feet
pin_no	varchar2(6)	feet
pin_location	varchar2(30)	
pin_elevation	varchar2(14)	feet
weather	varchar2(30)	feet
hi	varchar2(14)	

Neutron Probe Tables

Table monitoring_well_lab

well#	number(1)
depth	feet number(3)
sample _density	pcf number(4,1)
sample _moisture	% number(4,1)
passing _no200	% number(2)
liquid _limit	% number(2)
plasticity _index	% number(2)
note	varchar2(20)

Table Neutron_well_1

Well_date	ddmmmyyyy date
depth	feet number(3)
density	pcf number(4,1)
moisture	% number(4,1)
note	varchar2(20)

Table Neutron_well_2

well_date	ddmmmyyyy date
depth	feet number(3)
density	pcf number(4,1)
moisture	% number(4,1)
note	varchar2(20)

Laboratory Tests

Table lab_test_results

material	test_date	sampling	name_parameter	value_parameter
varchar2(25)	ddmmmyyyy varchar2(11)	varchar2(20)	varchar2(30)	varchar2(15)

Pavement Information

Table Modulus_FWD_backcalculation

test_date	layer#	material_type	modulus	poisson_ration	thick
date	number(1)	varchar2(30)	psi number(8,1)	number(3,2)	inch number(5,2)

5.1.5 Group 5, FWD Tests—Summary.

FWD data is stored in two tables, **FWD_parameter** and **FWD_test. FWD_parameter** contains one row for each FWD test sequence. **FWD_test** contains one row for each drop in a test sequence. The FWD tables are not related to any of the other tables in the database except through testdate in **FWD_parameter** and time in **FWD_test**. A separate description of the procedures for relating FWD test data to dynamic sensor response data is given later.

5.1.5.1 Table Structures—Group 5, FWD Tests.

Table FWD parameter

test#	integer primary key	
testdate	number(8) not null YYYYMMDD	
ndrops	number(2) not null	
totaldrops	number(2) not null	
nsensors	number(1) not null	
radius	number(4,2) not null	inch
loc1	number(3,1) not null	inch
loc2	number(3,1)	inch
loc3	number(3,1)	inch
loc4	number(3,1)	inch
loc5	number(3,1)	inch
loc6	number(3,1)	inch
loc7	number(3,1)	inch
note	varchar2(1000)	
test_type	varchar2(20)	

Table FWD_test

test#	integer	references FWD_parameter (test#)
station	number(2) not null	
lane	number(2) not null	
temperature	number(4,1) not null	Degrees F.
time	number(4) not null HHMM	
loadp	number(6) not null	lb
defl1	number(2,1) not null	mils
defl2	number(2,1)	mils
defl3	number(2,1)	mils
defl4	number(2,1)	mils
defl5	number(2,1)	mils
defl6	number(2,1)	mils
defl7	number(2,1)	mils
north_34rw	number(7,3)	ft
east_34rw	number(7,3)	ft
slab#	varchar2(3)	
north_slab	number(5,3)	ft
east_slab	number(5,3)	ft
heading	varchar2(8)	

Note: The database structure described in this section was accurate at the time of writing. But minor details of the structure are subject to change and should be checked if incompatibilities are apparent when using the database.

5.1.6 Processing and Storing FWD Test Data.

FWD tests are run periodically with simultaneous recording of the dynamic sensor responses during the impact time period. The dynamic sensor computer program described earlier is used to process the responses except that the procedure for selecting sensors for data storage is slightly different. The sensor response data is also stored in the database in the same way as for an aircraft event except that some of the entries in the **aircraft** table have different meanings.

During processing of the dynamic sensor responses, the ratio of the peak value to the RMS of the noise for each sensor is calculated with a smoothing filter cutoff frequency of 70 Hz. If the ratio is less than a preset threshold (40 for H-Bar and Carlson and 50 for LVDT sensors) the sensor responses are not stored because the sensor responses are in the noise level or the peak values are too low. The remaining sensors are selected for further processing. The peak values are then recalculated with a filter cutoff frequency of 200 Hz and are stored in the database for the selected sensors. Peak records (complete time histories) are stored in the **event_records** table if the position of the sensor is at the FWD drop position.

The processed data is stored in the database as an "FWD event" rather than an "aircraft event" by entering "FWD" in the aircraft_type column of the **aircraft** table. FWD data can therefore be retrieved with suitable queries on the **aircraft** table. Other changes in the meanings of the aircraft table entries are shown below. Otherwise, the data storage is exactly the same as for an aircraft event.

Table aircraft

event#	number(38) primary key	same as aircraft event
datetime	char(11) YYYYDDDhhmm	same as aircraft event
seconds	number(2)	same as aircraft event
aircraft#	varchar2(20) not null	FWD machine name
aircraftweight	number(7) lb	target load (kips)
carrier	varchar2(20) lb	type of FWD test
NvalidIR	number(1)	drop number
v0	number(6,3) ft/s	lane number
accel	number(7,3) ft/s^2	station number (test number)
yposition	number(7,3) ft	N/A
note	varchar2(1000)	names of gages over which the FWD was dropped
acpicture	varchar2(50)	N/A
filename	varchar2(20)	same as aircraft event
aircraft_type	varchar2(20)	FWD

Entries in the heading column in table **FWD_test** are automatically filled with default descriptions. The defaults are as follows:

- For load transfer tests, the default heading goes across the joint which is nearest to the FWD drop position.
- For all other tests, the default headings are along the test lanes. If the direction of the lane is East-West, the default heading is West. If the direction of the lane is North-South, the default heading is North.

A method to link the deflection data in the FWD tables with the dynamic sensor response data is:

a. Find "FWD" in the **aircraft** table and read the data for that row from the **aircraft** table.

b. Select the relative data from the sensor response tables and the **event_records** table where *event#* has the same value as in the aircraft table.

c. Select *test#* and others from the **FWD_parameter** table where *testdate* and *test_type* are respectively equal to the date in *datetime* and the type of FWD test in the **aircraft** table.

d. Select deflections and others from the **FWD_test** table where *test#* has the same value as in **FWD_parameter**, the station and the lane are respectively equal to those in the **aircraft** table, and the load value is close to the target load in the **aircraft** table.

<u>Example 1</u>

A user finds that the value in the *aircraft_type* column is FWD and wants to view the test parameters and the surface deflections from the FWD test. Assume that the values in the **aircraft** table:

event# = 92   datetime = 19960630746   target load = 50(kips),
drop No = 2   laneNo. = 5   station No. = 6
type of FWD test = Gage Verification.

The test date is the 63rd day in 1996, i.e., on March 3, 1996.

After entering SQLPLUS, the user can follow the steps below:

(1) **select * from fwd_parameter**
where testdate=960303 and test_type='Gage Verification';

```
TEST#  TESTDATE  NDROPS  TOTALDROPS  NSENSORS RADIUS LOC1 LOC2 LOC3 LOC4 LOC5 LOC6 LOC7 NOTE  TEST_TYPE
----- -------- ------ ---------- -------- ------ ---- ---- ---- ---- ---- ---- ---- ---- ---------
 25    960303    3     135       7      5.9    0   12   24   36   48   60   72       Gage Verification
```

(2) **select * from fwd_test**
where test#=25 and station=6 and lane=5 and 60000>loadp and loadp>45000;

```
TEST#  STATION  LANE  TEMPERATURE  TIME  LOADP  DEFL1  DEFL2  DEFL3  DEFL4  DEFL5  DEFL6  DEFL7  NORTH_34RW
EAST_34RW SLAB NORTH_SLAB EAST_SLAB
----- ------- ---- ----------- ---- ----- ----- ----- ----- ----- ----- ----- ----- ---------- --------- ---- ---------- ---------
 25    6    5    18    749 55455  7.8   7.4   7.7   6.1   5.4   4.7   4.1    270    58.417   D2    10    2.167
 25    6    5    18    749 55232  7.7   7.3   7.5   6.1   5.4   4.8   4.1    270    58.417   D2    10    2.167
 25    6    5    18    749 55057  7.7   7.3   7.6   6.2   5.4   4.8   4.1    270    58.417   D2    10    2.167
```

Because drop No = 2, the 2nd row is the result.

<u>Example 2</u>

Assume that the values below are found from the "FWD" row in the **aircraft** table:

event# = 88   datetime = 1996063060439   target load = 25(kips),
drop No = 2   laneNo. = 7   station No. = 4
the type of the FWD test = Load transfer.

The test date is the 63rd day in 1996, i.e., on March 3, 1996.

After entering SQLPLUS, the user can follow the steps below:

(1) **select * from fwd_parameter**
where testdate=960303 and test_type= 'Load Transfer';

TEST#	TESTDATE	NDROPS	TOTALDROPS	NSENSORS	RADIUS	LOC1	LOC2	LOC3	LOC4	LOC5	LOC6	LOC7	NOTE	TEST_TYPE
26	960303	6	78	7	5.9	0	12	24	36	48	60	72		Load Transfer

(2) **select * from fwd_test**
where test#=26 and station=4 and lane=7 and 35000>loadp and loadp>23000;

TEST#	STATION	LANE	TEMPERATURE	TIME	LOADP	DEFL1	DEFL2	DEFL3	DEFL4	DEFL5	DEFL6	DEFL7	NORTH_34RW	EAST_34RW	SLAB	NORTH_SLAB	EAST_SLAB
26	4	7	18	604	30753	9.4	2.6	2.4	2.2	2	1.9	1.7	281	62.25	D3	1	6
26	4	7	18	604	31198	9.3	2.5	2.3	2.1	2	1.8	1.7	281	62.25	D3	1	6
26	4	7	18	604	31436	9.4	2.6	2.4	2.1	2	1.8	1.7	281	62.25	D3	1	6

Because drop No = 2, the 2nd row in the above three rows is the result.

<u>Example 3</u>
Assume that the values below are found from the “FWD” row in aircraft table:

event# = 85   datetime = 19960621107   target load = 50(kips),
drop No = 2   laneNo. = 8   station No. = 2
the type of the FWD test = Reference.

The test date is the 62nd day in 1996, i.e., on March 2, 1996. The hour is 11 o’clock (GMT).

After entering SQLPLUS, the user can follow the steps below:
(1) **select * from fwd_parameter**
where testdate=960302 and test_type='Reference';

TEST#	TESTDATE	NDROPS	TOTALDROPS	NSENSORS	RADIUS	LOC1	LOC2	LOC3	LOC4	LOC5	LOC6	LOC7	NOTE	TEST_TYPE
21	960302	3	12	7	5.9	0	12	24	36	48	60	72		Reference
22	960302	3	12	7	5.9	0	12	24	36	48	60	72		Reference

The above two rows indicate that two groups of the FWD reference tests were conducted on March 2, 1996. The test time should be different in the two tests.

(2) **select * from fwd_test**
where test#= &test_number and station=2 and lane=8
and trunc(time,-2)=1100 and loadp<60000 and loadp>45000;

Execute the above SQL statements and respond with 21 to the following prompts:

Enter value for &test_number: 21

old   2: where test#= &test_number and station=2 and lane=8 and trunc(time,-2)=1100
new   2: where test#=21 and station=2 and lane=8 and trunc(time,-2)=1100
no rows selected

Execute the SQL statements again and respond with 22 to the following prompts:
Enter value for &test_number: 22
old   2: where test#= &test_number and station=2 and lane=8 and trunc(time,-2)=1100
new   2: where test#=22 and station=2 and lane=8 and trunc(time,-2)=1100

```
TEST# STATION LANE TEMPERATURE TIME LOADP DEFL1 DEFL2 DEFL3 DEFL4 DEFL5 DEFL6 DEFL7 NORTH_34RW
EAST_34RW SLAB NORTH_SLAB EAST_SLAB
----- ------- ---- ----------- ---- ----- ----- ----- ----- ----- ----- ----- ----- ---------- --------- ---- ---------- ---------
22   2  8  12  1108   56408  7.4   6   5.6   5   4.6   4    3.6   290   65.25   D3   10   9
22   2  8  12  1108   56455  7.4   6   5.5   5   4.5   4    3.5   290   65.25   D3   10   9
22   2  8  12  1108   56551  7.4   6   5.6   5   4.5   4.1  3.5   290   65.25   D3   10   9
```

Because drop No = 2, the 2nd row is the result.

6. REFERENCE.

1. William T. Thompson, *Theory of Vibrations With Applications,* 2nd Edition, Prentice-Hall, Inc., 1981.